BAREKENG: Journal of Mathematics and Its Applications
September 2026 Volume 20 Issue 3 Page 2631-2644
P-ISSN: 1978-7227 E-ISSN: 2615-3017
https://doi.org/10.30598/barekengvol20iss3pp2631-2644

A COMPARATIVE ANALYSIS OF COLOR SPACES FOR TOMATO RIPENESS CLASSIFICATION USING MACHINE LEARNING AND DEEP LEARNING APPROACHES

Firda Fadri [1*], **Yoyok Yulianto** [2], **Kiswara Agung Santoso** [3]

[1,2,3]*Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Jember*
Jln. Kalimantan No.37, Jember, 68121, Indonesia

*Corresponding author's e-mail: * **firdafadri@unej.ac.id***

Article Info

Article History:

Received: 7th October 2025
Revised: 2nd December 2025
Accepted: 17th March 2026
Published: 8th April 2026

Keywords:

Classification;
Color space;
Deep Learning;
Machine Learning;
Tomato ripeness.

ABSTRACT

The classification of tomato ripeness is crucial for post-harvest processing, quality assurance, and agricultural automation, as manual evaluation is often subjective, inconsistent, and time-consuming. This research investigated the impact of color space selection and hyperparameter optimization on tomato ripeness classification using machine learning (SVM, Random Forest, K-NN, GNB) and deep learning (CNN) approaches. Evaluation results indicated that YCbCr was the best-performing color space for classical models, with SVM achieving the highest accuracy (91.24%) and RF following closely (89.54%), whereas HSV yielded optimal performance for CNN (90.46%), highlighting differences in feature extraction mechanisms. Confusion matrix and ROC curve analyses demonstrated that models capturing nonlinear and interdependent color features, such as SVMs and CNNs, achieved superior class separability, particularly for the Ripe and Unripe classes. Dominant channel analysis revealed that chrominance channels, Cb in YCbCr and H in HSV, played a critical role in ripeness discrimination. These findings emphasized the importance of preprocessing for feature selection and provided guidance on selecting appropriate models and color spaces to improve the accuracy and reliability of automated tomato ripeness classification.

How to cite this article:

F. Fadri, Y. Yulianto and K. A. Santoso., "A COMPARATIVE ANALYSIS OF COLOR SPACES FOR TOMATO RIPENESS CLASSIFICATION USING MACHINE LEARNING AND DEEP LEARNING APPROACHES", *BAREKENG: J. Math. & App.,* vol. 20, no. 3, pp. 2631-2644, Sep, 2026.

Journal homepage: *https://ojs3.unpatti.ac.id/index.php/barekeng/*
Journal e-mail: *barekeng.math@yahoo.com; barekeng.journal@mail.unpatti.ac.id*

Research Article · Open Access

1. INTRODUCTION

Tomatoes had a crucial role in the agricultural supply chain, both consumed directly and utilised as raw materials for processed industries. The level of ripeness greatly affected the flavour, shelf life, and selling price, so a method of determining ripeness is needed [1], [2]. Ripeness tests were generally divided into two types, destructive and non-destructive method [3]. Destructive methods were usually conducted through the analysis of chemical content or physical characteristics, such as sugar content, acidity, or texture. Non-destructive methods, such as digital image analysis allowed measurements to be made without damaging the sample, making them more suitable for application in the distribution chain [4]. Ripeness was usually checked by workers looking at the fruit, but this method can be unreliable, slow, and expensive. A non-destructive system was needed that uses technology to automatically, quickly, and reliably check tomato ripeness [5], [6].

The development of image processing technology has enabled the application of machine learning approaches in tomato ripeness classification. By using digital images, visual patterns that were difficult to interpret consistently by human vision can be analyzed more objectively. This non-destructive method helped check quality more accurately and lowers the chance of losses from harvesting at the wrong time or damage during distribution [7], [8]. Common methods included Support Vector Machine (SVM), Random Forest (RF), K-Nearest Neighbors (K-NN), and Gaussian Naïve Bayes (GNB). These algorithms have been applied in various research. The Convolutional Neural Network (CNN) algorithm has emerged as a new standard in image processing due to its ability to automatically and hierarchically extract features without requiring manual feature engineering [9], [10], [11]. The use of these algorithms aimed to recognize visual patterns in images, such as differences in color, texture, and shape, to provide key information for determining tomato ripeness. Each method used a different approach and parameter selection for data processing, so classification results may vary depending on the characteristics of the dataset [12], [13].

When applied to multiclass classification problems such as tomato ripeness determination (unripe, ripe, overripe, and damaged), these machine learning algorithms offered several advantages. These methods generally distinguished classes well, reduced potential errors due to data complexity, and provided flexibility in handling variations in image characteristics. In CNN architectures specifically, the method proved highly effective at handling lighting variations and complex backgrounds in fruit images captured in uncontrolled environments [14]. In addition, some algorithms were efficient at processing large amounts of data, so they remain relevant for fruit image classification research [15].

In addition to algorithm selection, another factor influencing classification success was the color space used in image processing. Color was the main indicator of tomato ripeness, but how it is represented can vary depending on the chosen color space, such as RGB, HSV, CIE Lab, or YCbCr. The different color spaces produced variations in feature quality, so choosing the right color model is important to obtain more accurate classification results [16], [17]. This study also explored the identification of the dominant color channel within each color space. Analyzing dominant channels was important because some channels may carry more discriminative information about ripeness levels, such as changes in hue intensity or luminance. Identifying the most significant channel helped improve feature selection strategies, enhance model interpretability, and potentially reduce computational complexity by focusing on the most informative components.

Previous research had utilized image-based, non-destructive methods powered by machine learning for fruit classification, including mango, kiwi, avocado, tomato, and strawberry [18], [19], [20], [21], [22], [23]. Most research had employed only a single algorithm or single-color space, without conducting comprehensive performance comparisons across different methods and color spaces. Moreover, limited research had examined the role of dominant color channels within different color spaces in supporting classification accuracy. Therefore, a gap remained in understanding the combined impact of algorithm selection, color space variation, and dominant channel characteristics on classification performance. This research addressed that gap by evaluating several machine learning algorithms (SVM, RF, K-NN, and GNB) and a deep learning model (CNN) across multiple color spaces, while also analyzing dominant color

channels to determine the most optimal combination for tomato ripeness classification. The findings could contribute to the development of a more effective and transparent fruit image classification system to advance precision agriculture practices.

2. RESEARCH METHODS

2.1 Dataset

This dataset consisted of tomato images acquired in a laboratory environment. Images were captured with a smartphone camera, flash enabled, with each tomato placed against a white background and photographed at a fixed distance to maintain uniform scale. All images were recorded at a resolution of 4,624 × 3,472 pixels. Representative samples of each ripeness category are shown in Fig. 1.

Figure 1. Datasets Representing Each Fruit Ripeness Category

Tomato images were classified into four ripeness levels: Damaged, Overripe, Ripe, and Unripe. The dataset consisted of 6,511 training images and 765 test images. Table 1 shows the distribution of the training and testing data.

Table 1. Dataset Distribution

Distribution	Ripeness level			
	Damaged	**Overripe**	**Ripe**	**Unripe**
Train	959	1992	1975	1585
Test	106	232	230	197

***Data source**: Processed by the author (2026)*

2.2 Technical Research

The research began with image acquisition, where tomato samples were captured under controlled laboratory conditions to ensure consistent illumination. The preprocessing stage involved cropping the images to remove extraneous background regions, resizing them to 128 × 128 pixels for uniform input size, and converting them into four color spaces (RGB, HSV, CIE Lab, and YCbCr). Feature extraction was performed by calculating statistical descriptors (mean, standard deviation, skewness, and kurtosis) for each color channel to represent the visual characteristics of the ripeness levels.

Based on Table 1, the dataset was split into training and testing subsets using a 90:10 ratio to maximize model learning while maintaining a robust test set of 765 samples for evaluation. A class imbalance was observed among the ripeness categories, with minority classes such as 'Damaged' having fewer samples. To mitigate bias toward majority classes, weighted class balancing was applied during training by assigning class weights inversely proportional to class frequencies, enhancing fairness and robustness in classification.

The implementation phase involved five algorithms (SVM, RF, KNN, GNB, and CNN). Hyperparameter tuning was performed using GridSearchCV with stratified 5-fold cross-validation. Model performance was evaluated using accuracy, precision, recall, F1-score, and ROC-AUC metrics. A channel dominance analysis was conducted to identify the most discriminative components in each color space for tomato ripeness classification. The research workflow was illustrated in Fig. 2.

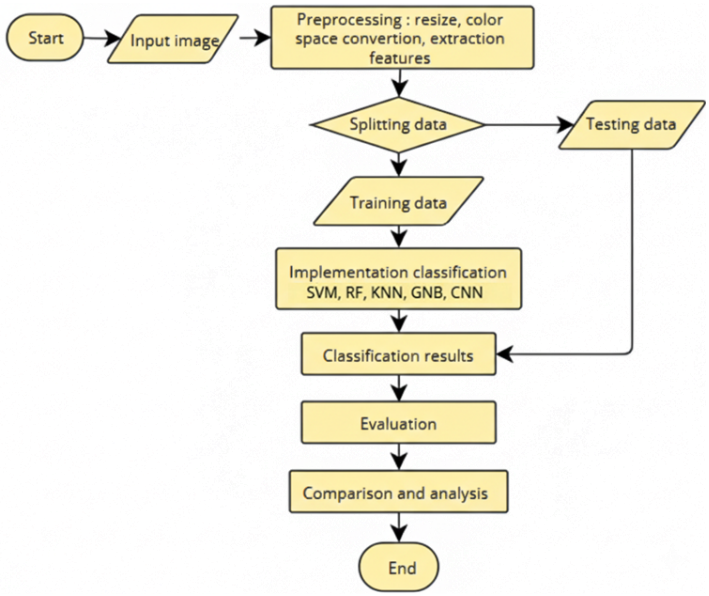

Figure 2. Research Workflow

2.3 Color Space

RGB (Red, Green, Blue) was the primary color space used in digital image processing [24]. In this color space, each pixel is defined by three color intensity components: red (R), green (G), and blue (B). By combining three channels, any color can be represented. Eq. (1) represented each pixel based on the R, G, and B (R,G,B) components, and Eq. (2) normalized these values to a range between 0 and 1.

$$P(x,y) = [R(x,y), G(x,y), B(x,y)], \tag{1}$$

$$R'(x,y) = \frac{R(x,y)}{255}, \quad G'(x,y) = \frac{G(x,y)}{255}, \quad B'(x,y) = \frac{B(x,y)}{255}. \tag{2}$$

The advantage of RGB is that its representation is intuitive and directly matches the output of display devices (monitors and cameras), making it easy to use in image preprocessing. However, it was sensitive to luminance and intensity, so differences in brightness could affect the R, G, and B values, even if the object's color remained the same [25]. For example, a ripe tomato and an overripe tomato might have color differences that were difficult to distinguish from RGB values alone if the lighting was inconsistent. Therefore, although RGB was effective for basic representation, transformation to other color spaces such as HSV, CIE Lab, or YCbCr was often performed to separate color information from intensity or to emphasise features relevant to tomato ripeness classification.

HSV (Hue, Saturation, Value) separated the hue, saturation, and brightness information of an image. Eq. (3) was used to find the maximum (C_{max}) and minimum (C_{min}) values of the normalized RGB components. Subsequently, the hue (H) was calculated using Eq. (4), the saturation (S) was determined using Eq. (5), and the value (V) is defined using Eq. (6).

$$C_{max} = \max(R', G', B'), \quad C_{min} = \min(R', G', B'), \quad \Delta = C_{mzx} - C_{min}, \tag{3}$$

$$H = \begin{cases} 0, & if\ \Delta = 0 \\ 60°.\left(\dfrac{G'-B'}{\Delta} mod 6\right), & if\ C_{max} = R' \\ 60°.\left(\dfrac{B'-R'}{\Delta} + 2\right), & if\ C_{max} = G'\ , \\ 60°.\left(\dfrac{R'-G'}{\Delta} + 4\right), & if\ C_{max} = B' \end{cases} \tag{4}$$

$$S = \begin{cases} 0, & if\ C_{max} = 0 \\ \dfrac{\Delta}{C_{max}}, & otherwise' \end{cases} \tag{5}$$

$$V = C_{max}. \tag{6}$$

CIE Lab (L*, a*, b*) was designed to mimic human color perception more accurately than device-based color models such as RGB. In this color space, the L* component reflected lightness, a* captured the green–red range, and b* showed the blue–yellow range. The main advantage of the Lab color space lies in its ability to emphasise subtle color differences, such as in distinguishing between ripe and overripe tomatoes, as small spectral differences could be more easily detected than in the RGB space.

The process of converting colors from RGB to CIE Lab involved passing through the XYZ intermediate color space. First, the values of each RGB channel were normalized to 0–1, and then gamma correction (linearization) was applied to compensate for the display's nonlinear characteristics. After that, the linear RGB values were converted to the XYZ color space using a standard transformation matrix with an illuminant of D65 as the white light reference, as shown in Eq. (7).

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix}. \tag{7}$$

Once the X, Y, and Z values were obtained, each component was normalised against the D65 reference white point, i.e. $X_n = 0.95047, Y_n = 1.00000$, and dan $Z_n = 1.08883$. The next step was the transformation from XYZ to CIE Lab using Eq. (8).

$$L^* = 116\left(\frac{Y}{Y_n}\right)^{1/3} - 16, \quad a^* = 500\left[\left(\frac{X}{X_n}\right)^{\frac{1}{3}} - \left(\frac{Y}{Y_n}\right)^{\frac{1}{3}}\right], \quad b^* = 200\left[\left(\frac{Y}{Y_n}\right)^{\frac{1}{3}} - \left(\frac{Z}{Z_n}\right)^{\frac{1}{3}}\right]. \tag{8}$$

This process transformed the device-dependent RGB color space into the universal CIE Lab color space, which corresponds to human visual perception. The CIE Lab color space was widely used in image processing, color analysis, and color-based object recognition.

YCbCr separated the luminance (Y) and chrominance (Cb, Cr) components, making it more stable against lighting variations. The transformation from RGB to YCbCr is used Eq. (9).

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.0168736 & -0.331264 & 0.5 \\ 0.5 & -0.418688 & -0.081312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}. \tag{9}$$

Implementing these color spaces facilitated the extraction of color features for the classification of tomato images. RGB served as the foundation. HSV and Lab highlighted color and brightness characteristics relevant to ripeness. YCbCr separates luminance from chrominance, making the model more resilient to lighting changes [26].

2.4 Machine Learning

Machine learning was utilized in image processing for tomato ripeness classification to construct models that can recognize patterns in color and texture. SVM, RF, K-NN, and GNB were commonly employed [3].

2.4.1 Support Vector Machine

SVM distinguished classes by identifying the hyperplane that maximized the margin between them. For datasets that were not linearly separable, a kernel function was applied to map the data into a higher-dimensional space, enabling linear separation. The SVM decision function was expressed as follows in Eq. (10).

$$f(x) = w.x + b, \tag{10}$$

where w was the weight vector, and b is the bias. The aim of SVM training was to minimize the objective function, $\min_{w,b} \frac{1}{2}\|w\|^2$, subject to the constrain $y_i(w.x_i + b) \geq 1, \forall i$. In cases where the data were non-linear case, the kernel $K(x_i, y_i)$ was employed to project the data to a high-dimensional feature space, as defined in Eq. (11).

$$f(x) = \sum_{i=1}^{n} a_i y_i K(x_i, x) + b, \tag{11}$$

where a_i were the Lagrange coefficients.

2.4.2 Random Forest

Random Forest (RF) is an ensemble learning technique that consists of multiple decision trees. Each tree was trained on a randomly selected subset of the dataset, called a bootstrap sample. Random selection of features was employed to improve generalization and mitigate overfitting. The final class prediction was determined through majority voting, as expressed in Eq. (12).

$$\hat{y} = mode\ \{h_1(x), h_2(x), \ldots, h_T(x)\}, \tag{12}$$

$h_t(x)$ was the prediction result of tree t. T is the total number of trees in the forest. The separation function at each tree node was determined by minimizing impurity using the Gini index criterion, using Eq. (13).

$$Gini = 1 - \sum_{k=1}^{K} p_k^2, \tag{13}$$

where p_k was the proportion of data in class k within the node.

2.4.3 K-Nearest Neighbors

K-Nearest Neighbors (K-NN) is a method that classifies new objects based on the K training data points closest to them. The most common distance used was the Euclidean Distance, as shown in Eq. (14).

$$d(x, x_i) = \sqrt{\sum_{j=1}^{n} (x_j - x_{ij})^2}. \tag{14}$$

The predicted class was determined by the majority of labels of the K nearest neighbors, as defined by (15).

$$\hat{y} = mode\ \{y_i | x_i \in N_K(x)\}. \tag{15}$$

$N_K(x)$ was the set of K nearest neighbors of the test data x.

2.4.4 Gaussian Naive Bayes

Gaussian Naive Bayes (GNB) was a probabilistic classifier based on Bayes' theorem that assumed each feature was conditionally independent of its class. The probability that data $x = (x_1, x_2, \ldots, x_n)$ belonged to class C_k was calculated using the following Eq. (16).

$$P(C_k|x) = \frac{P(C_k)\prod_{i=1}^{n} P(x_i|C_k)}{P(x)}. \quad (16)$$

For continuous data, the distribution of $P(x_i|C_k)$ in Eq. (17) was assumed to follow a Gaussian (normal) distribution.

$$P(x_i|C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} exp\left(-\frac{(x_i - \mu_{k,i})^2}{2\sigma_{k,i}^2}\right), \quad (17)$$

$\mu_{k,i}$ represented the mean and $\sigma_{k,i}$ represented standard deviation of feature x_i belonging to class C_k.

2.5 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) were a type of deep learning architecture designed to handle grid-like data structures, including digital images. Unlike conventional machine learning approaches that relied on manually crafted features, CNNs automatically extracted hierarchical representations directly from raw pixel data, capturing simple patterns such as edges in the initial layers and progressively more complex structures in deeper layers [27]. The fundamental operation in a CNN was convolution, in which a $k \times k$ filter (kernel) moves across the input image to generate a feature map. The convolution operation was expressed in Eq. (18).

$$y_{i,j} = \sigma\left(\sum_{m=0}^{k-1}\sum_{n=0}^{k-1} w_{m,n} \cdot x_{i+m,j+n} + b\right), \quad (18)$$

where $y_{i,j}$ was output value at position (i,j) in feature map, $x_{i+m,j+n}$ was the input value at position $(i+m, j+n)$, $w_{m,n}$ represented the kernel weight at position (m,n), b was the bias term, and $\sigma(\cdot)$ denoted the nonlinear activation function, usually ReLU ($f(z) = \max(0,z)$).

After convolution, a pooling operation (typically Max Pooling) was applied to reduce the spatial dimensions of the feature maps, thereby speeding up computation and reducing overfitting. The extracted features were then flattened and passed to a Fully Connected Layer to produce the final classification probabilities using the Softmax function, as defined Eq. (19).

$$P(y = c|\mathrm{x}) = \frac{e^{z_c}}{\sum_{j=1}^{C} e^{z_j}}, \quad (19)$$

where $P(y = c|\mathrm{x})$ represented the predicted probability that the input x belonged to class c, and c is the total number of classes, and z_c denoted the output (logit) of the Fully Connected Layer for class c, and e was the base of the natural logarithm.

The proposed CNN architecture in this research received input images of size $128 \times 128 \times 3$, representing the spatial resolution and three-channel color information. The network comprised three convolutional layers with 32, 64, and 128 filters, respectively, each using a 3×3 kernel and ReLU activation function. To progressively reduce spatial dimensions and enhance feature abstraction, each convolutional layer was followed by a 2×2 max-pooling layer. This hierarchical structure enabled the extraction of low-level features such as edges and textures in the early layers, while capturing more complex ripeness-related patterns in the deeper layers. The resulting feature maps were flattened and connected to a fully connected layer with 128 neurons, followed by a dropout layer (0.3 or 0.5) to mitigate overfitting. The final layer consisted of four neurons with Softmax activation to classify tomato images into four ripeness categories. Hyperparameter tuning was conducted on batch size (32, 64), learning rate (1e−3, 1e−4), and dropout rate to ensure stable convergence and optimal performance across color spaces.

2.6 Confusion Matrix

A confusion matrix is a metric used to evaluate the performance of classification models. It indicates the number of correct and incorrect predictions made by the model for each class. Table 2 presents the confusion matrix.

Table 2. Confusion Matrix

Actual/ Predicted	Positive	Negative
Positive	True positive (TP)	False negative (FN)
Negative	False positive (FP)	True negative (TN)

The confusion matrix provides the basis for calculating evaluation metrics, including accuracy, precision, recall, and F1-score. These metrics are computed using Eqs. (20), (21), (22), and (23), respectively.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{20}$$

$$Precision = \frac{TP}{TP + FP}, \tag{21}$$

$$Recall = \frac{TP}{TP + FN}, \tag{22}$$

$$F1 - score = \frac{2\times(Precision\times Recall)}{Precision+Recall}. \tag{23}$$

3. RESULTS AND DISCUSSION

Optimal hyperparameter configurations influenced model performance in both algorithmic complexity and color space representation. The best hyperparameter configurations presented in Table 3 showed that SVM consistently selected the RBF kernel with a high C value (100), indicating the need for nonlinear decision boundaries, while moderate gamma values maintained a balanced bias–variance trade-off. RF relied on a relatively large number of trees (200–500) with controlled tree depth, emphasizing the importance of ensemble strength in capturing inter-channel variations. K-NN exhibited stable settings (k = 5, distance weighting), suggesting that local similarity remained informative across color spaces. GNB retained identical var_smoothing values, reflecting limited sensitivity to color transformation. Variations in CNN batch size and dropout rate across color spaces indicated that deep feature extraction was responsive to differences in color representation structure.

The evaluation results in Table 4 demonstrated that color space selection significantly affected classification performance. SVM achieved the highest overall accuracy (91.24%) in the YCbCr color space, followed by RF (89.54%) and K-NN (88.24%) in the same color space, indicating strong generalization and balanced classification across ripeness categories. The superior performance in YCbCr suggested that separating the luminance (Y) and chrominance (Cb, Cr) components reduced illumination effects while preserving discriminative color information. CNN achieved the best performance in the HSV color space (90.46%), indicating that deep feature extraction responded differently to color representation than classical models. GNB consistently yielded the lowest performance across all color spaces, indicating that its independence assumption failed to capture the nonlinear color distributions in ripeness classification.

The difference in the best-performing color space between classical machine learning and deep learning models was likely due to their distinct feature-extraction mechanisms. Classical algorithms relied on predefined statistical color features, making them more sensitive to how effectively a color space separated discriminative information. CNN performed hierarchical feature extraction through convolutional layers, enabling it to learn complex spatial–color patterns directly from the data. CNN performance was

less dependent on initial channel separation, which explained why HSV could outperform other color spaces in the deep learning approach. This contrast highlighted that the effectiveness of color space depended on the learning paradigm used.

Table 3. Best Hyperparameter Configuration

Method	Parameter	Range of value	Color space			
			RGB	HSV	CIE Lab	YCbCr
SVM	C	[0.1, 1, 10, 100]	100	100	100	100
	gamma	['scale', 'auto', 0.001, 0.01, 0.1, 1]	0.1	0.1	scale	0.1
	kernel	['rbf', 'poly']	rbf	rbf	rbf	rbf
RF	n_estimators	[100, 200, 500]	500	500	200	200
	max_depth	[None, 10, 20, 30]	None	20	30	20
	min_sample_split	[2, 5, 10]	2	2	2	2
	min_sample_leaf	[1, 2, 4]	1	1	1	1
K-NN	n_neighbors	[3, 5, 7, 9, 11]	5	5	5	3
	weights	['uniform', 'distance']	'distance'	'distance'	'distance'	'distance'
	metric	['euclidian', 'manhattan', 'minkowski']	'euclidean'	'manhattan'	'euclidean'	'euclidean'
GNB	var_smoothing	[1e-10, 5e-10, 1e-9, 5e-9, 1e-8]	1e-10	1e-10	1e-10	1e-10
CNN	batch size	[32,64]	32	64	32	64
	learning rate	[1e-3, 1e-4]	1e-3	1e-3	1e-3	1e-3
	dropout rate	[0.3, 0.5]	0.5	0.3	0.5	0.3

***Data source**: Processed by the author (2026)*

Table 4. Evaluation Results

Method	Metric	Color space			
		RGB	HSV	CIE Lab	YCbCr
SVM	Accuracy (%)	85.23	85.62	90.20	**91.24**
	Precision (%)	82.78	83.63	88.31	89.62
	Recall (%)	83.50	84.26	89.16	90.53
	F1-score (%)	82.92	83.79	88.68	90.02
RF	Accuracy (%)	86.14	86.01	88.89	**89.54**
	Precision (%)	83.71	84.09	86.28	87.45
	Recall (%)	81.84	82.36	84.93	86.11
	F1-score (%)	82.42	82.98	85.38	86.58
K-NN	Accuracy (%)	83.92	81.44	87.19	**88.24**
	Precision (%)	82.80	79.37	85.45	86.34
	Recall (%)	79.96	78.13	83.76	84.81
	F1-score (%)	80.85	78.59	84.35	85.38
GNB	Accuracy (%)	46.67	48.63	64.31	**66.80**
	Precision (%)	46.19	46.50	59.83	61.20
	Recall (%)	42.68	43.58	58.76	60.31
	F1-score (%)	41.87	42.95	58.72	60.02
CNN	Accuracy (%)	89.54	**90.46**	87.45	87.32
	Precision (%)	87.99	88.77	86.01	84.89
	Recall (%)	87.71	87.90	83.84	85.24
	F1-score (%)	87.73	88.26	84.63	84.91

***Data source**: Processed by the author (2026)*

The matrices for the SVM, RF, K-NN, GNB, and CNN classifiers are shown in Fig. 3. The SVM model Fig. 3 (a) demonstrated the most balanced classification performance, with relatively few misclassifications across categories. The RF Fig. 3 (b) and K-NN Fig. 3 (c) models also maintained high accuracy for the Ripe and Unripe classes but exhibited greater confusion in the Damaged category, with a notable number of samples misclassified as Overripe. The GNB model (Figure 3d) yielded the weakest performance, with substantially lower diagonal values and pronounced misclassification, particularly 56

Ripe samples predicted as Overripe and 75 Damaged samples also classified as Overripe. The CNN model Fig. 3 (e) achieved the highest number of correctly classified Ripe instances, indicating strong capability to recognize dominant ripeness characteristics; however, its performance on the Damaged class was slightly lower than that of SVM, suggesting a minor trade-off in class-specific sensitivity. These findings indicated that models capable of capturing nonlinear and interdependent color features, such as SVM, achieved superior class separability in the YCbCr and CNN in HSV color spaces. The dominance of diagonal elements in the SVM and CNN matrices confirmed their robustness and reliability in differentiating tomato ripeness levels, whereas the probabilistic assumptions of GNB limited its adaptability to overlapping color distributions.

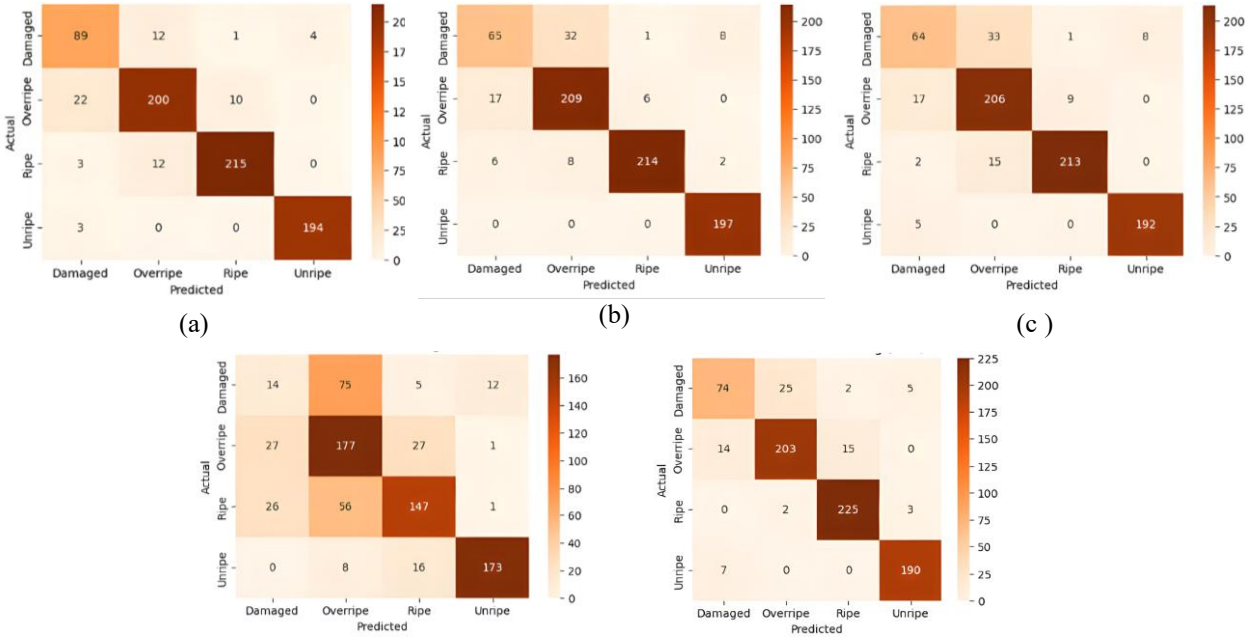

Figure 3. Confusion Matrices for Best Color Space
(a) SVM, (b) RF, (c) K-NN, (d) GNB, (e) CNN

ROC curves were presented in Fig. 4, illustrating the trade-off between sensitivity and specificity for each classifier in their best-performing color spaces. The machine learning models (SVM, RF, K-NN, and GNB) achieved peak performance in the YCbCr space. The SVM model Fig. 4 (a) achieved the highest Macro-average AUC (0.9825) and Micro-average AUC (0.9853), demonstrating consistent and near-perfect classification across all classes, particularly for Ripe (AUC=0.9954) and Unripe (AUC=0.9986) samples. The RF Fig. 4 (b) and K-NN Fig. 4 (c) models also performed well in YCbCr, with slightly higher Micro-average than Macro-average AUCs, indicating stronger performance on majority classes (Ripe and Unripe) but relatively lower separability for the minority Damaged class. GNB Fig. 4 (d) recorded the lowest Macro-average (0.8101) and Micro-average (0.8515) AUCs, largely due to its inability to distinguish Damaged samples (AUC=0.5792), reflecting the limitations of its probabilistic assumptions with overlapping color distributions.

The CNN model inFig. 4 (e) achieved optimal performance in HSV, with a macro-average AUC of 0.9826 and micro-average AUC of 0.9868, indicating strong overall performance and balanced class predictions. This shift from YCbCr for ML models to HSV for CNN suggested that deep learning could leverage richer spatial and color information, whereas traditional ML relied on the luminance-chrominance separation of YCbCr to handle nonlinear color patterns. The comparisons of micro and macro-average AUCs highlighted that SVM in YCbCr and CNN in HSV were the most robust approaches for tomato ripeness detection, with SVM achieving balanced class separability and CNN exploiting complex feature representations for superior performance across all classes.

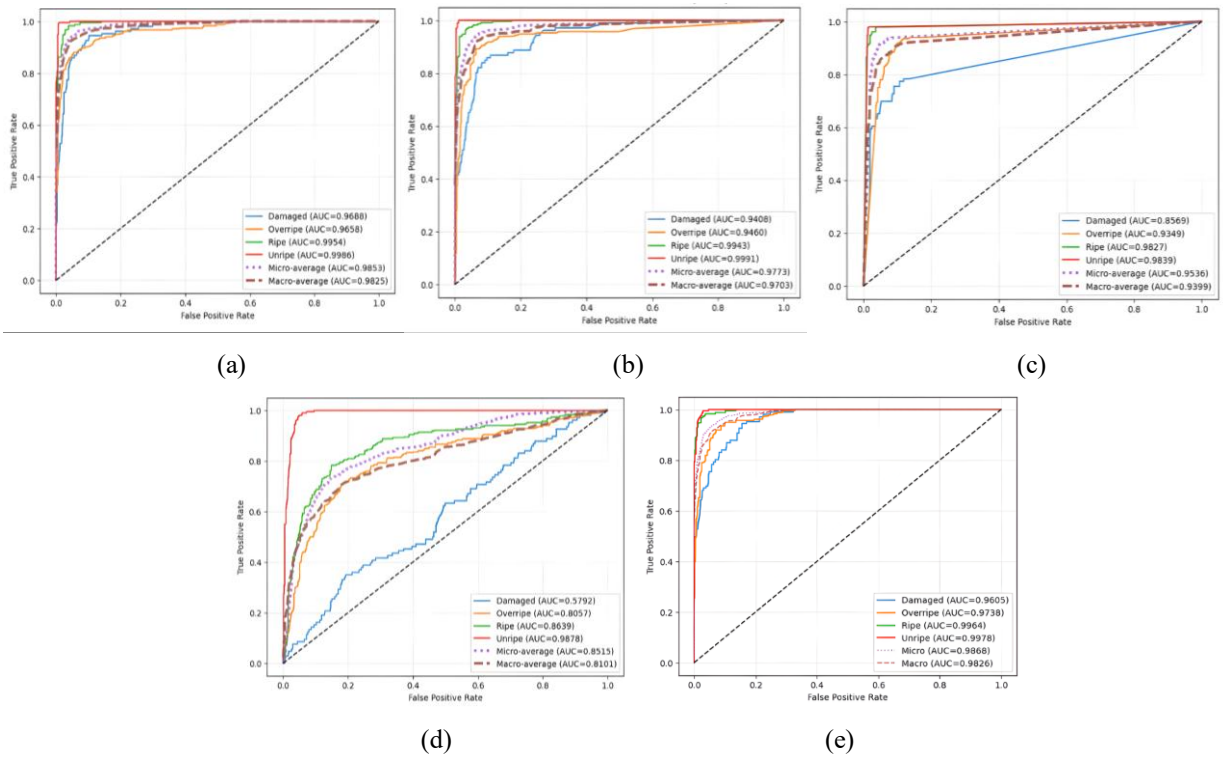

Figure 4. ROC Curve for Best Color Space
(a) SVM, (b) RF, (c) K-NN, (d) GNB, (e) CNN

Following the ROC curve evaluation, Table 5 presents the contributions of dominant color channels in the best-performing color spaces across different classification methods. The analysis revealed that chrominance channels, particularly Cb in YCbCr and H in HSV, played a crucial role in distinguishing fruit ripeness. These findings emphasized the importance of identifying dominant channels to optimize feature selection and improve classification accuracy.

Table 5. Dominant Color Channel Analysis

Method	Best color space	Channel	Contribution (%)
SVM	YCbCr	Y	28.15%
		Cb	**38.11%**
		Cr	33.74%
RF	YCbCr	Y	17.23%
		Cb	**55.43%**
		Cr	27.34%
K-NN	YCbCr	Y	**38.08%**
		Cb	34.31%
		Cr	27.61%
GNB	YCbCr	Y	**38.08%**
		Cb	34.31%
		Cr	27.61%
CNN	HSV	H	**43.68%**
		S	37.45%
		V	18.86%

***Data source**: Processed by the author (2026)*

For SVM, the Cb channel contributed the most (38.11%), followed by Cr (33.74%) and Y (28.15%), indicating that color difference information was more informative than luminance. Similarly, RF showed a strong dominance of the Cb channel (55.43%), suggesting that ensemble-based methods effectively leveraged chrominance variations. In contrast, K-NN and GNB exhibited a more balanced pattern, with the Y channel slightly ahead (38.08%), suggesting that brightness information remained relevant across distance-based and probabilistic models. For CNN, the HSV color space was used, with the H channel dominating (43.68%), indicating that deep learning effectively captured hue variations, which were essential for ripeness detection. These results underscore the importance of checking dominant color channels during preprocessing, as focusing on the most informative channels optimized feature selection, reduced computational complexity, and enhanced classification performance by improving the discrimination of fruit ripeness.

4. CONCLUSION

The results of the research demonstrated that color space selection and hyperparameter optimization significantly influenced the performance of tomato ripeness classification. Classical machine learning models performed best in YCbCr, leveraging the separation of luminance and chrominance, while CNN, as a deep learning model, achieved optimal results in HSV by capturing hierarchical spatial–color features. Chrominance channels, particularly Cb for traditional models and H for CNN, were critical for discriminating ripeness, whereas brightness remained relevant for distance-based and probabilistic models. Models capable of capturing nonlinear and interdependent color information, such as SVM and CNN, achieved superior accuracy and class separability. Identifying dominant color channels during preprocessing enhanced feature selection, reduced computational complexity, and improved classification reliability, providing a robust framework for automated fruit ripeness detection.

Author Contributions

Firda Fadri: Conceptualization, Funding Acquisition, Data Curation, Formal Analysis, Writing-Original Draft, Writing-Review and Editing, Validation. Yoyok Yulianto: Resources, Data Collection, Software, Writing-Review, and Editing. Kiswara Agung Santoso: Methodology, Formal Analysis, Writing-Original Draft, Validation. All authors discussed the results and contributed to the final manuscript.

Funding Statement

This research was financially supported by the Beginner Lecturer Research Grant, Institute for Research and Community Service, University of Jember.

Acknowledgment

This research was financially supported by the Beginner Lecturer Research Grant, Institute for Research and Community Service, University of Jember.

Declarations

The authors declare no conflicts of interest to report.

Declaration of Generative AI and AI-assisted technologies

Generative AI tools (e.g., ChatGPT) were used solely for language refinement (grammar, spelling, and clarity). The scientific content, analysis, interpretation, and conclusions were developed entirely by the authors. The authors reviewed and approved all final text.

REFERENCES

[1] T. Khatun, A. Razzak, and S. Islam, "AN EXTENSIVE REAL-WORLD IN FIELD TOMATO IMAGE DATASET INVOLVING MATURITY CLASSIFICATION AND RECOGNITION OF FRESH AND DEFECT TOMATOES," *Data Br.*, vol. 51, p. 109688, 2023, doi: https://doi.org/10.1016/j.dib.2023.109688

[2] R. Shinoda, H. Kataoka, K. Hara, and R. Noguchi, "SMART AGRICULTURAL TECHNOLOGY TRANSFORMER-BASED RIPENESS SEGMENTATION FOR TOMATOES," *Smart Agric. Technol.*, vol. 4, no. February, p. 100196, 2023, doi: https://doi.org/10.1016/j.atech.2023.100196 .

[3] M. Islam, S. Bijjahalli, T. Fahey, and A. Gardi, *DESTRUCTIVE AND NON - DESTRUCTIVE MEASUREMENT APPROACHES AND THE APPLICATION OF AI MODELS IN PRECISION AGRICULTURE : a review*, vol. 25, no. 3. Springer US, 2024.doi: https://doi.org/10.1007/s11119-024-10112-5

[4] M. Yazdani, D. Bao, J. Zhou, A. Wang, and R. D. van Klinken, "SINGLE-WAVELENGTH NEAR-INFRARED IMAGING AND MACHINE LEARNING FOR DETECTING QUEENSLAND FRUIT FLY DAMAGE IN CHERRIES," *Smart Agric. Technol.*, vol. 12, p. 101090, 2025, doi: https://doi.org/10.1016/j.atech.2025.101090.

[5] R. Khodabakhshian and M. H. Abbaspour-Fard, "PATTERN RECOGNITION-BASED RAMAN SPECTROSCOPY FOR NON-DESTRUCTIVE DETECTION OF POMEGRANATES DURING MATURITY," *Spectrochim. Acta Part A Mol. Biomol. Spectrosc.*, vol. 231, p. 118127, 2020, doi: https://doi.org/10.1016/j.saa.2020.118127.

[6] Q. Wang *et al.*, "IN SITU NONDESTRUCTIVE IDENTIFICATION OF CITRUS FRUIT RIPENESS VIA HYPERSPECTRAL IMAGING TECHNOLOGY," *Plant Methods*, 2025, doi: https://doi.org/10.1186/s13007-025-01354-z

[7] S. Sikder, M. S. Islam, M. Islam, and S. Reza, "MACHINE LEARNING WITH APPLICATIONS IMPROVING MANGO RIPENESS GRADING ACCURACY : A COMPREHENSIVE ANALYSIS OF DEEP LEARNING , TRADITIONAL MACHINE LEARNING , AND TRANSFER LEARNING TECHNIQUES," *Mach. Learn. with Appl.*, vol. 19, no. May 2024, p. 100619, 2025, doi: https://doi.org/10.1016/j.mlwa.2025.100619

[8] O.-R. Rusu *et al.*, "InteRACTION EFFECTS OF CULTIVARS AND NUTRITION ON QUALITY AND YIELD OF TOMATO," *Horticulturae*, vol. 9, no. 5, 2023, doi: https://doi.org/10.3390/horticulturae9050541 .

[9] H. S. Mputu, A. Abdel-mawgood, and A. Shimada, "TOMATO QUALITY CLASSIFICATION BASED ON TRANSFER LEARNING FEATURE EXTRACTION AND MACHINE LEARNING ALGORITHM CLASSIFIERS," *IEEE Access*, vol. 12, no. December 2023, pp. 8283–8295, 2024, doi:. https://doi.org/10.1109/ACCESS.2024.3352745

[10] R. Qasrawi, M. Amro, R. Zaghal, M. Sawafteh, and S. V. Polo, "MACHINE LEARNING TECHNIQUES FOR TOMATO PLANT DISEASES CLUSTERING, PREDICTION AND CLASSIFICATION," in *2021 International Conference on Promising Electronic Technologies (ICPET)*, 2021, pp. 40–45, doi: https://doi.org/10.1109/ICPET53277.2021.00014.

[11] A. Sanga and V. Zsófia, "EVALUATION OF MACHINE LEARNING ALGORITHM'S SELECTION CRITERIA FOR SATELLITE IMAGE CLASSIFICATION, LEVERAGING ACCURACY AND EXECUTION TIMES," *Egypt. J. Remote Sens. Sp. Sci.*, vol. 28, no. 3, pp. 534–541, 2025, doi: https://doi.org/10.1016/j.ejrs.2025.08.001

[12] H. Azarmdel, A. Jahanbakhshi, S. S. Mohtasebi, and A. R. Muñoz, "EVALUATION OF IMAGE PROCESSING TECHNIQUE AS AN EXPERT SYSTEM IN MULBERRY FRUIT GRADING BASED ON RIPENESS LEVEL USING ARTIFICIAL NEURAL NETWORKS (ANNS) AND SUPPORT VECTOR MACHINE (SVM)," *Postharvest Biol. Technol.*, vol. 166, p. 111201, 2020, doi: https://doi.org/10.1016/j.postharvbio.2020.111201

[13] A. A. Adam and R. Alfred, "OPTIMIZING THE CLASSIFICATION PERFORMANCE BY FINE-TUNING THE MACHINE LEARNING HYPERPARAMETERS AND UTILIZING PCA AND RFE FEATURE SELECTION METHODS," in *Proceedings of the 4th International Conference on Advances in Computational Science and Engineering*, 2024, pp. 435–454.doi: https://doi.org/10.1007/978-981-97-2977-7_27

[14] J. Chen *et al.*, "DETECTING RIPE FRUITS UNDER NATURAL OCCLUSION AND ILLUMINATION CONDITIONS," *Comput. Electron. Agric.*, vol. 190, p. 106450, 2021, doi: https://doi.org/10.1016/j.compag.2021.106450

[15] M. Knott, F. Perez-Cruz, and T. Defraeye, "FACILITATED MACHINE LEARNING FOR IMAGE-BASED FRUIT QUALITY ASSESSMENT." 2022, doi: https://doi.org/10.1016/j.jfoodeng.2022.111401

[16] G. C. Wakchaure *et al.*, "MATURITY STAGES DETECTION PROTOTYPE DEVICE FOR CLASSIFYING CUSTARD APPLE (ANNONA SQUAMOSA L) FRUIT USING IMAGE PROCESSING APPROACH," *Smart Agric. Technol.*, vol. 7, p. 100394, 2024, doi: https://doi.org/10.1016/j.atech.2023.100394.

[17] C. Pestana, N. Akhtar, W. Liu, D. Glance, and A. Mian, "ADVERSARIAL ATTACKS AND DEFENSE ON DEEP LEARNING CLASSIFICATION MODELS USING YCBCR COLOR IMAGES," in *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1–9, doi: https://doi.org/10.1109/IJCNN52387.2021.9533495.

[18] D. Worasawate, P. Sakunasinha, and S. Chiangga, "AUTOMATIC CLASSIFICATION OF THE RIPENESS STAGE OF MANGO FRUIT USING A MACHINE LEARNING APPROACH," *AgriEngineering*, vol. 4, no. 1, pp. 32–47, 2022, doi: https://doi.org/10.3390/agriengineering4010003

[19] A. Raghavendra, D. S. Guru, M. K. Rao, and R. Sumithra, "ARTIFICIAL INTELLIGENCE IN AGRICULTURE HIERARCHICAL APPROACH FOR RIPENESS GRADING OF MANGOES," *Artif. Intell. Agric.*, vol. 4, pp. 243–252, 2020, doi:. https://doi.org/10.1016/j.aiia.2020.10.003

[20] K. Chawgien and S. Kiattisin, "MACHINE LEARNING TECHNIQUES FOR CLASSIFYING THE SWEETNESS OF WATERMELON USING ACOUSTIC SIGNAL AND IMAGE PROCESSING," *Comput. Electron. Agric.*, vol. 181, p. 105938, 2021, doi: https://doi.org/10.1016/j.compag.2020.105938

[21] C. Huang *et al.*, "STRUCTURED HYPERSPECTRAL IMAGING AND MACHINE LEARNING FOR NON-DESTRUCTIVE KIWIFRUIT FIRMNESS PREDICTION, CLASSIFICATION, AND INTELLIGENT POST-

HARVEST MANAGEMENT," *J. Food Compos. Anal.*, vol. 147, p. 108026, 2025, doi: https://doi.org/10.1016/j.jfca.2025.108026

[22] P. Tipauksorn, P. Luekhong, M. Okada, J. Thongpron, C. Nadee, and K. Yingkayun, "OPTIMIZED ENSEMBLE LEARNING FOR NON-DESTRUCTIVE AVOCADO RIPENESS CLASSIFICATION," *Smart Agric. Technol.*, vol. 12, p. 101114, 2025, doi: https://doi.org/10.1016/j.atech.2025.101114.

[23] A. N. De Oliveira *et al.*, "TOMATO CLASSIFICATION USING MASS SPECTROMETRY-MACHINE LEARNING TECHNIQUE : A FOOD SAFETY-ENHANCING PLATFORM," *Food Chem.*, vol. 398, no. August 2022, p. 133870, 2023, doi: https://doi.org/10.1016/j.foodchem.2022.133870.

[24] S. Alharthi and A. Gutub, "ADJUSTING IMAGE STEGO PRACTICALITY VIA YCBCR COLOR SPACE FORMATION," *J. Eng. Res.*, no. January, 2025, doi: https://doi.org/10.1016/j.jer.2025.07.008.

[25] R. C. Gonzalez and R. E. Woods, *DIGITAL IMAGE PROCESSING*. Pearson, 2018.

[26] R. Lukac and K.N. Plataniotis, *COLOR IMAGE PROCESSING: METHODS AND APPLICATIONS (1st ed.)*. CRC Press, 2006, doi: https://doi.org/10.1201/9781420009781

[27] E. Xhaferra, E. Cina, and L. Toti, "ClaSSIFICATION OF STANDARD FASHION MNIST DATASET USING DEEP LEARNING BASED CNN ALGORITHMS," in *2022 International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 2022, pp. 494–498, doi: https://doi.org/10.1109/ISMSIT56059.2022.9932737