Matrik: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer
Vol. 23, No. 3, July 2024, pp. 677∼689
ISSN: 2476-9843, accredited by Kemenristekdikti, Decree No: 200/M/KPT/2020
DOI: 10.30812/matrik.v23i3.3823

Color Feature Extraction for Grape Variety Identification: Naïve Bayes Approach

Dolly Indra , Fitriyani Umar , Putri Jafar
Universitas Muslim Indonesia, Makassar, Indonesia

Article Info

Article history:

Received January 25, 2024
Revised May 07, 2024
Accepted July 08, 2024

Keywords:

Color Feature Extraction
Grape Variety Identification
Naïve Bayes

ABSTRACT

The problem addressed in this research is the lack of an efficient and accurate method for automatically identifying grape varieties. Accurate identification is crucial for quality control in the agricultural and food industries, impacting product labeling, pricing, and consumer trust. **The aim of this research** is to develop an automated system to classify green, black, and red grapes using digital image processing technology. **This research method** employs Naïve Bayes classification combined with color feature extraction. Testing was conducted under two scenarios: a database scenario with predefined grape image datasets and an out-of-database scenario with images resembling grape colors. Image processing includes resizing images to 200 × 200 pixels, Gamma Correction, Gaussian filtering, conversion to Lab* color space, K-Means Clustering for segmentation, followed by feature extraction and Naïve Bayes classification. **The results of this** research are that in the database scenario, the system achieved accuracies of 98.33% with an 80:20 data split and 98.89% with a 70:30 split. In the out-of-database scenario, accuracies were 96.67% with an 80:20 split and 97.78% with a 70:30 split. The conclusion of this research is the proposed method provides a reliable and efficient solution for automatic grape variety identification, benefiting quality control in agriculture and food industries.

Corresponding Author:

Dolly Indra, +6281343720253
Faculty of Computer Science,
Universitas Muslim Indonesia, Makassar, Indonesia.
Email: dolly.indra@umi.ac.id

How to Cite:
P. Jafar, D. Indra, and F. Umar, "Color Feature Extraction for Grape Variety Identification: Naïve Bayes Approach", *MATRIK: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer*, Vol. 23, No. 3, pp. 677-689, July, 2024.

1. INTRODUCTION

The rapid advancement of digital innovation in information and communication technology in this era has transformed every sector and industry through digital transformation [1]. Technological advancements, particularly in pattern recognition for object identification, have become a major aspect influencing the evolution of teaching and learning methods in the current digital era [2]. Classifying grape types based on color feature extraction is an intriguing problem in the field of digital image processing. Identifying grape varieties is highly relevant in the agricultural industry, grape sales, and scientific research related to health and nutrition. The problem lies in the lack of an efficient and accurate method for automatically identifying grape varieties, which currently relies on manual observation and is prone to human error. This problem needs to be resolved to ensure quality control, accurate product labeling, and appropriate pricing and to maintain consumer trust. Understanding grape varieties can be enhanced by applying technology that automatically classifies grape types. Although each individual may have unique perspectives and judgments regarding these objects, differences in opinion regarding grape types can be overcome through the automation approach of technology.

Research [3] utilized KNN and SVM separately to classify grape leaves as healthy or unhealthy based on texture and color features. The evaluation was conducted by testing both methods, where KNN with K=1 achieved an accuracy of 96.66%, while SVM yielded an accuracy of 90%. Study [4] focused on texture feature extraction in grape images and classification using the K-Nearest Neighbor algorithm. The steps involved in this research included converting images from RGB to grayscale, GLCM feature analysis, GLCM feature calculation, and GLCM feature extraction results. The highest classification result reached an accuracy of 99.5441% with k = 2 at level 8. Research [5] utilized edge image processing and geometric morphology. The grape detection process was based on concave point detection and optimal contour segment clustering. Steps in this research included using bilateral filters and the Canny algorithm. Evaluation results showed that the average precision rate for various grape types was 87.76%, and the average recall rate was 90.04%.

There are gaps that have not been resolved by previous research, namely the specific identification of grape types based on color feature extraction using an efficient and accurate automated method. Most prior studies focused on other features like texture or size or used different classifiers like KNN and SVM, which, while effective, did not specifically address the need for an automated color-based classification system for grape types. **The difference between** this research and the previous one is its focus on grape-type identification based on color feature extraction using the Naïve Bayes method. Naïve Bayes is employed to calculate the probability of hypothesis classes by considering their attributes and determining the class with the highest probability [6]. Naïve Bayes excels in operating with relatively small training datasets, efficiently selecting estimated parameters during the classification process [7]. Although previous studies have utilized different approaches such as KNN, SVM, or texture extraction methods, this research specifically addresses the classification problem of grape types based on color. In image processing, feature extraction plays a crucial role and is an essential step in pattern recognition. These features represent precise information extracted from segmented characters (symbols or words), enabling clear differentiation between characters [8]. This research also utilizes pre-processing methods such as Gamma correction, Gaussian filtering, and K-Means Clustering segmentation as steps before the extraction phase. Thus, this study contributes to expanding the scope of existing research by introducing a new approach that focuses more on grape-type identification.

This study aims to classify grape types based on their color characteristics using an effective method. This research's significance lies in its potential to reduce human error, ensure consistency, and improve efficiency in grape production by automating the identification process. Using Naïve Bayes is particularly advantageous due to its robustness and effectiveness with small datasets, providing reliable results with minimal computational resources. Advanced image processing techniques like Gamma correction, Gaussian filtering, and K-Means Clustering enhance the system's accuracy under various image conditions. This novel approach builds on existing methodologies and introduces new technological applications in viticulture, leading to better management practices, improved crop yields, and more precise grape plant genetics research. **The study contributes** to the field by introducing a probabilistic approach to grape classification, expanding methodologies in agricultural image processing

2. RESEARCH METHOD

The method employed in this research is quantitative. Quantitative methodology prioritizes collecting numerical data and statistical analysis to address research questions and test hypotheses. This study uses numerical data to analyze color features from grape images and classify grapes into specific categories using the Nave Bayes algorithm. The proposed system in this research is represented as a block diagram, which can be seen in Figure 1.

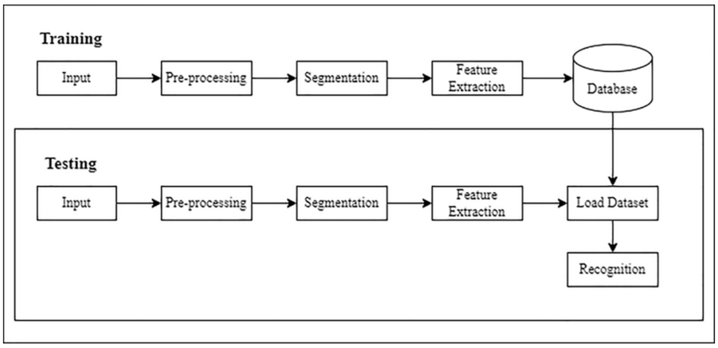


Figure 1. Proposed System

2.1. Input

The input of grape fruit test images includes varieties such as shine muscat for green grapes, black autumn for black grapes, and red globe for red grapes. These images have been resized to 200 $\times$ 200 pixels using an application and captured manually with an iPhone 11 smartphone equipped with a 12 MP camera (wide and ultra-wide). Each image is taken per fruit using natural outdoor light, with the image capture distance set at 10 cm to ensure optimal accuracy and detail.

2.2. Pre-Processing

Pre-processing is the initial stage in machine learning, where data is transformed or encoded to enable efficient analysis by machines [9]. Pre-processing is a crucial step in various image applications, enhancing image quality by removing noise and reducing distortions. In this stage, we perform several steps, including Gamma correction, Gaussian filtering, and color conversion. Gamma correction is a non-linear operation on original image pixels, modifying pixel values to enhance image quality by leveraging the projection relationship between pixel values and gamma values according to an internal mapping [10]. Gamma correction aims to improve image quality by adjusting contrast. In this stage, we use Gamma correction with a Gamma value of 1.2 to provide contrast to the image. Gaussian filtering is a method used to filter images more effectively, smoothing or blurring images and removing noise [11]. In this stage, we use a Gaussian filter with a sigma value of 1.0, allowing us to adjust the extent to which Gaussian filtering effects are applied according to the image's characteristics. Color conversion is the process of changing the color of an object to enhance the ability to identify color information more intuitively. The color space conversion we use in the pre-processing stage converts from RGB color to L*a*b* color space. The Lab color space enhances color distribution and is capable of representing all colors visible to the human eye [12].

2.3. Segmentation

Segmentation is a pixel-level classification task where an image consists of various pixels that, when grouped together, determine different elements in the image [13]. In this stage, we use segmentation, specifically the K-Means Clustering algorithm. K-means clustering performs the segmentation process by determining the desired number of clusters. The result is the cluster indices and cluster centers. Pixel labels are reconstructed from cluster indices, and the segmented image is obtained. Subsequently, objects within each cluster are represented by cells, and clusters containing objects (foreground) are identified based on their area. Segmentation is performed by creating a binary image showing only the foreground clusters, filling holes, and removing unwanted small objects. Finally, the segmented image is displayed as an RGB image. The K-Means Clustering algorithm can be described by Equation 1. Every digital image waiting to be segmented can be considered as a set of data points $X = \{X_1, X_2, \ldots, X_n\}$ with an n-dimensional vector. By using the K-Means segmentation, the image is divided into K groups. The normal method involves finding a subset $Z = \{C_1, C_2, \ldots, C_k\}$ within the set X to minimize the objective function $J = \sum_{i=1}^{k} \sum_{xj\epsilon ci}$. Where $d_{ij}(X_j, C_i)$ is the Euclidean distance from the data point X_j to the cluster center C_i [14].

$$J = \sum_{i-1}^{k} \sum_{xj\epsilon ci} \shortmid\mid X_j - C_i \mid\shortmid^2 \tag{1}$$

2.4. Feature Extraction

Feature extraction is a method in data processing that involves extracting new features from the original dataset to reduce the amount of resources needed for processing. In this stage, we use color feature extraction, where the method or process involves extracting information from the color components of an image or picture using descriptive color statistics such as mean and standard deviation. The mean provides a measure of the average value of the dataset and is commonly used in various statistical analyses to understand the overall data distribution. The standard deviation measures the amount of variation or dispersion in a dataset. It quantifies how much the values in the dataset deviate from the mean. A low standard deviation indicates that the values are close to the mean, whereas a high standard deviation indicates that the values are spread out over a wider range. The extraction process begins with the color image being divided into three main channels: red, green, and blue. Next, the mean and standard deviation of pixel intensities are calculated for each color channel. The formulas for mean and standard deviation are shown in Equations 2 and Equations 3 [15].

$$\bar{x} = \frac{\sum_i^N |f(i) - \bar{x}|^2}{N-1} \tag{2}$$

$\bar{x}$ is a symbol that represents the mean (average) of the dataset. $\sum_i^N f(i)$ is the summation of the values from $i = 1$ to $i = N$. $f(i)$ represents the value of the *i*-th observation in the dataset. N is the total number of observations in the dataset. σ is a symbol that represents the standard deviation of the dataset. $\sum_i^N f(i)$ is the summation of the values from $i = 1$ to $i = N$. $f(i)$ represents the value of the i-th observation in the dataset. $\bar{x}$ is the mean (average) of the dataset. N is the total number of observations in the dataset.

$$\sigma = \sqrt{\frac{\sum_i^N \mid f(i) - \bar{x} \mid^2}{N-1}} \tag{3}$$

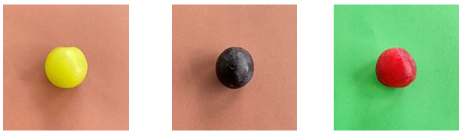
2.5. Recognition

Recognition is an effort to identify and localize objects within a predefined environment context accurately [16]. In the recognition stage, we perform classification using the Naïve Bayes method. The Naïve Bayes Classification Algorithm is a simple method for building a classification model, relying on Bayes' Theorem of conditional probability and assuming strong independence [17]. This research uses the Naïve Bayes algorithm to classify grape types based on color features. Expected outcomes include high accuracy and precision in identifying grape varieties, benefiting agricultural practices, and comparing performance with other classification methods. The study contributes to the field by introducing a probabilistic approach to grape classification, expanding methodologies in agricultural image processing. Bayes' theorem equation can be described in Equation 4 [18]. P(H|E) is the final probability of the conditional probability of a hypothesis H occurring if evidence E occurs. P(E|H) is the probability that evidence E occurring will affect hypothesis H. P(H) is the initial (prior) probability that hypothesis H will occur without considering any evidence. P(E) is the initial (prior) probability of evidence E occurring without considering any other hypotheses/evidence.

$$P(H \mid E) = \frac{D(E \mid H).P(H)}{P(E)} \tag{4}$$

3. RESULT AND ANALYSIS

This research employs two data testing scenarios: an 80:20 and a 70:30 testing scenario, using 300 images in RGB format with JPG file extension. Additionally, testing scenarios outside the database were conducted, along with elaboration on the implementation of the developed application. Testing for this system was conducted under two scenarios: database scenario and out-of-database scenario. The database scenario utilized grape image datasets, including shine muscat grapes for green grapes, black autumn grapes for black grapes, and red globe grapes for red grapes. On the other hand, out-of-database testing utilized manually collected image databases resembling the colors of green, black, and red grapes. Images in the database scenario employed shine muscat grapes for green grapes, black autumn grapes for black grapes, and red globe grapes for red grapes. The dataset used in this stage was obtained by capturing images independently using an iPhone 11 smartphone camera with a 12 MP camera specification. It involved capturing outdoor images from morning to noon, between 9 a.m. to 12 p.m. This process utilized natural outdoor light, with the image capture distance set at approximately 10 cm to ensure optimal accuracy and detail. The overall dataset consisted of 300 images divided into 210 images for training data and 90 for testing data. Some of the data obtained from this data collection are shown in Figure 2.



(a) green grapes (b) black grapes (c) red grapes

Figure 2. Grape Images

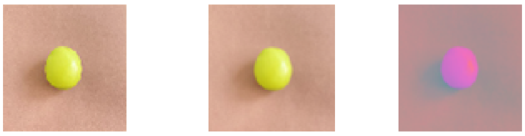
3.1. Input Process

In this image input stage, the process involves extracting images from the previously created dataset and initializing them as original images. This initial step is crucial as it sets the foundation for further image processing and analysis. Below is an example of image input in Figure 3. This figure illustrates the original images as they appear right after extraction, showcasing their raw form before any preprocessing steps are applied.

Figure 3. Original Image Input

3.2. Pre-Processing

In the pre-processing stage, several steps are carried out. Starting with the first stage, a correction process is performed using Gamma correction with a Gamma value of 1.2. Then, the resulting image that has undergone Gamma correction will undergo filtering using a Gaussian filter with a sigma of 1.0. The filtered image will then be converted by transforming the RGB image into an L*a*b* image, which will be used in the segmentation process. The pre-processing results can be seen in Figure 4.



(a) Gamma Correction (b) Filter Gaussian (c) L*a*b* Image

Figure 4. Pre-Processing

3.3. Segmentation

The segmentation process aims to separate the object and background from the grape image. The segmentation method used is K-Means Clustering segmentation, dividing the segmented image into three clusters using Euclidean squared distance and three replications, dividing the image based on the area size for each cluster, resulting in cluster indices and cluster centers. Each cluster will represent the number of pixels in the image classified in that cluster. The selected segmentation result is the cluster containing the foreground separated from the image's background. The segmentation result produces an image with RGB colors. The outcome of the segmentation using K-Means clustering can be seen in Figure 5. K-means clustering produces segmentation results by separating the object, which is the foreground, from the image's background. The background, which originally had color, changes to black, while the object retains its original color.

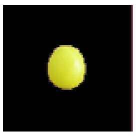


Figure 5. Segmentation

3.4. Features Extraction

In this stage, feature extraction is performed by extracting information from the color components of an image using descriptive color statistics, namely mean and standard deviation, from the RGB color components. The calculated values of color feature extraction for the training data images are computed using two data split scenarios: an 80:20 split and a 70:30 split from 300 images. The color feature extraction values for the training data, with an 80:20 split and a 70:30 split, are presented in Table 1 and Table 2, respectively.

Table 1. Color Feature Extraction Values of 240 Training Data Images from 80:20 Data Split

Image.jpg	Mean			Standard Deviation		
	R	G	B	R	G	B
Green 1	188.017	192.095	624.032	617.008	631.564	226.894
Green 2	201.441	210.304	811.117	618.189	647.076	270.058
...	...	...	...	...	...	...
Green 79	174.706	178.120	673.435	583.707	596.804	249.597
Green 80	176.871	187.181	564.760	582.886	618.298	212.928
Black 1	550.570	446.865	469.495	227.819	200.424	209.727
Black 2	657.117	565.012	567.192	254.580	230.419	234.081
...	...	...	...	...	...	...
Black 79	615.662	548.770	557.047	241.377	222.631	228.440
Black 80	751.407	653.312	662.992	269.107	246.506	252.180
Red 1	124.308	479.842	429.630	453.913	197.174	184.169
Red 2	138.767	567.355	509.287	507.121	230.051	214.945
...	...	...	...	...	...	...
Red 79	166.422	913.532	670.677	576.526	346.303	272.857
Red 80	6.376.125	206.382	135.797	337.966	114.453	822.434

Table 2. Color Feature Extraction Values of 210 Training Data Images from 70:30 Data Split

Image.jpg	Mean			Standard Deviation		
	R	G	B	R	G	B
Green 1	188.017	192.095	624.032	617.008	631.564	226.894
Green 2	201.441	210.304	811.117	618.189	647.076	270.058
...	...	...	...	...	...	...
Green 69	174.968	182.220	613.647	583.658	609.185	230.637
Green 70	176.717	187.045	563.882	582.662	618.142	212.710
Black 1	565.957	457.015	478.560	231.220	202.409	211.264
Black 2	649.485	559.885	562.895	252.983	229.503	233.402
...	...	...	...	...	...	...
Black 69	596.592	512.900	535.560	237.643	214.810	222.190
Black 70	766.540	663.405	671.680	272.112	248.284	253.565
Red 1	124.308	479.842	429.630	453.913	197.174	184.169
Red 2	138.767	567.355	509.287	507.121	230.051	214.945
...	...	...	...	...	...	...
Red 69	165.740	918.480	742.025	560.344	328.058	282.454
Red 70	637.612	206.382	135.797	337.966	114.453	822.434

3.5. Testing Phase

Testing in this system is conducted with two testing scenarios: the database scenario and the out-of-database scenario. The database scenario involves using a dataset of images of green, black, and red grapes. Meanwhile, for out-of-database testing, a manually collected image database resembling the colors of green, black, and red grapes is used. Testing with ”Correct” results indicates accurate classification of grape types based on color, while ”Incorrect” results, highlighted in red, represent misclassifications based on color.

a. Scenario 1

The first testing scenario is carried out with an 80:20 dataset split from 300 images, using 60 test images, resulting in an accuracy of 98.33%. The testing results with an 80:20 data split ratio are shown in Table 3. Based on Table 3, each image went through the classification process in which, during the testing with 60 test images, one image was not recognized by the system, specifically the Red grape image 16, where the system's detection result was "Black." The testing results yielded an accuracy of:

$$\begin{aligned} Accuracy &= \frac{\textit{Number Of Correct Data}}{\textit{Total Number Of Data}} \times 100\% \\ &= \frac{59}{60} \times 100\% \\ &= 98,33\% \end{aligned}$$

Table 3. Color Feature Extraction Values of 60 Test Images from 80:20 Data Split

Image.jpg	Mean			Standard Deviation			Image Test	System Detection Result Image
	R	G	B	R	G	B		
Green 1	171.243	178.202	617.132	574.733	599.523	230.904	'Green'	'Green'
Green 2	170.720	175.714	589.465	579.400	598.226	228.632	'Green'	'Green'
...	...	...	...	...	...	...	...	...
Green 19	186.668	187.383	714.107	610.944	615.370	259.744	'Green'	'Green'
Green 20	188.993	191.612	598.595	614.228	624.587	227.901	'Green'	'Green'
Black 1	671.295	594.550	601.240	250.017	228.566	231.837	'Black'	'Black'
Black 2	301.607	256.695	257.992	176.276	153.254	154.223	'Black'	'Black'
...	...	...	...	...	...	...	...	...
Black 19	695.452	600.412	609.210	259.555	236.020	241.744	'Black'	'Black'
Black 20	611.055	477.787	504.460	229.469	200.510	211.151	'Black'	'Black'
Red 1	143.669	562.562	441.842	494.377	222.379	180.869	'Red'	'Red'
Red 2	160.029	735.510	604.085	551.855	279.630	245.250	'Red'	'Red'
...	...	...	...	...	...	...	...	...
Red 16	**537.955**	**380.990**	**371.572**	**325.811**	**233.730**	**233.023**	**'Red'**	**'Black'**
...	...	...	...	...	...	...	...	...
Red 19	130.915	588.110	471.507	467.382	249.339	211.567	'Red'	'Red'
Red 20	166.342	915.590	670.707	576.189	346.744	273.039	'Red'	'Red'

b. Scenario 2

The second testing scenario was conducted with a 70:30 dataset split from 300 images, using 90 test images, with an accuracy result of 98.89%. The testing results using the 70:30 data ratio are shown in Table 4. Based on Table 4, each image went through the classification process in which, during the testing with 90 test images, one image was not recognized by the system, specifically the Red grape image 7, where the system's detection result was "Black." The testing results yielded an accuracy of:

$$\begin{aligned} Accuracy &= \frac{\textit{Number Of Correct Data}}{\textit{Total Number Of Data}} \times 100\% \\ &= \frac{89}{90} \times 100\% \\ &= 98,89\% \end{aligned}$$

Table 4. Color Feature Extraction Values for 90 Test Images From 70:30 Data Split

Image.jpg	Mean			Standard Deviation			Image Test	System Detection Result Image
	R	G	B	R	G	B		
Green 1	203.402	207.000	587.075	633.193	646.725	203.325	'Green'	'Green'
Green 2	174.706	178.120	673.435	583.707	596.804	249.597	'Green'	'Green'
...	...	...	...	...	...	...	...	...
Green 29	191.587	192.293	629.847	624.608	628.597	230.912	'Green'	'Green'
Green 30	186.031	190.333	676.977	608.240	623.788	237.746	'Green'	'Green'
Black 1	577.167	521.437	588.235	224.302	209.183	233.855	'Black'	'Black'
Black 2	633.910	562.065	568.625	246.022	225.451	230.581	'Black'	'Black'
...	...	...	...	...	...	...	...	...
Black 29	679.720	595.357	595.262	254.698	230.895	232.515	'Black'	'Black'
Black 30	749.015	661.992	656.212	273.782	251.628	251.793	'Black'	'Black'
Red 1	135.496	670.490	590.245	473.839	257.171	232.293	'Red'	'Red'
Red 2	166.422	913.532	670.677	576.526	346.303	272.857	'Red'	'Red'
...	...	...	...	...	...	...	...	...
Red 7	**537.955**	**380.990**	**371.572**	**325.811**	**233.730**	**233.023**	**'Red'**	**'Black'**
...	...	...	...	...	...	...	...	...
Red 29	137.232	617.057	508.602	485.226	240.817	212.344	'Red'	'Red'
Red 30	163.000	645.072	579.722	546.451	236.394	220.775	'Red'	'Red'

c. Test Results Diagram Chart
The feature extraction results conducted by the authors are presented in average values for each color component R, G, and B in terms of mean and standard deviation to observe the differences between the features in the extraction results. In this stage, a bar chart diagram will be displayed, illustrating the averages of each color component (R, G, and B) for mean and standard deviation values during the testing process. The diagram of the test results for scenario 1 with an 80:20 data ratio and scenario 2 with a 70:30 data ratio is shown in Figure 6 and Figure 7.

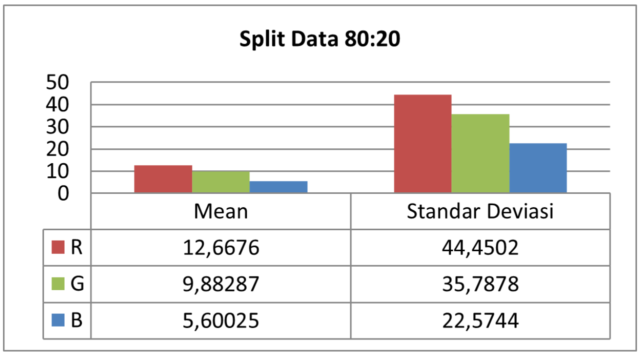


Figure 6. Testing Data Diagram Scenario 1

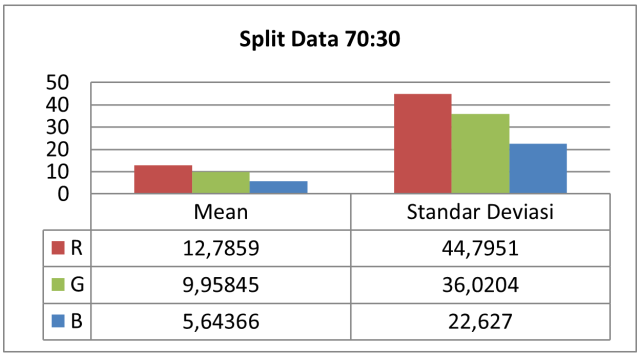


Figure 7. Testing Data Diagram Scenario 2

Based on the diagrams in Figure 6 and Figure 7, the R (Red) color component has the highest average mean and standard deviation values. The R (Red) color component has the highest mean, indicating that the red color has a dominant contribution to the image. The high standard deviation indicates a large variation in red color intensity across all images. The larger the standard deviation value, the greater the color variation present in the image. Thus, these results show that the red color plays a significant role and has a large variation in the image based on the RGB color feature extraction results.

d. Overall Test Results

The following are the overall test results for the entire data split based on the mean and standard deviation values conducted three times each for each data scenario: the 80:20 scenario, the 70:30 scenario, and the 60:40 scenario. The overall test results are shown in Table 5. Based on Table 5, it can be seen that the average obtained from the accuracy of the test data overall is 98.61%, and the average precision of the test data overall is 98.42%. The grape variety recognition test results show the highest accuracy level in the 70:30 data split scenario, producing an accuracy of 98.8% and a precision of 98.5%.

Table 5. Overall Test Results

No	Data Distribution Scenario	Test Data Accuracy (%)	Test Data Precision (%)
1	80:20	98,33	98,33
2	70:30	98,89	98,52
	Average	98,61	98,42

e. Out-of-Database Testing Scenario

The following are the results of testing out-of-database data with pictures of jujube fruit as green grapes, dates as black grapes, and cherry tomatoes as red grapes. The mean and standard deviation values were calculated three times for each data scenario: the 80:20 scenario, the 70:30 scenario, and the 60:40 scenario. The results of out-of-database testing are shown in Table 6. Based on Table 6, it can be seen that the overall accuracy obtained from out-of-database testing is 97.22%, with an average precision of 97.41%. The out-of-database testing results show that the highest accuracy is achieved with a 70:30 data split, reaching 97.78%, detecting out-of-database images as authentic and with a precision of 98.15%.

Table 6. Out-of-Database Testing Results

No	Data Distribution Scenario	Accuracy (%)	Precision (%)
1	80:20	96,67	96,67
2	70:30	97,78	98,15
	Average	97,22	97,41

f. Research Finding

The findings of this research are that using two testing scenarios, namely the 80:20 and 70:30 testing scenarios, with 300 RGB images in JPG file format, yielded high accuracy results in classifying grape varieties. The data preprocessing, including gamma correction, Gaussian filtering, color conversion, segmentation, and feature extraction, is crucial in preparing datasets suitable for machine learning algorithms. The results show that the 70:30 data split scenario achieved the highest accuracy of 98.89% and precision of 98.52%, indicating the effectiveness of the methodology used. The results of this research are in line with or supported by several previous studies that focused on grape classification based on texture and color features. For example, one study utilized KNN and SVM methods to classify grape leaves, achieving 96.66% accuracy with KNN and 90% with SVM. This supports the high accuracy findings of this research, though this research demonstrates even higher accuracy using more complex image processing methods. Another study employed the KNN method to classify grape images based on texture features, with the highest accuracy result of 99.5441%. Although this study's focus was different, it underscores the importance of texture and color features in grape image classification, aligning with the findings of this research. Furthermore, a study on grape berry detection and size measurement based on edge image processing, though with different applications, supports the use of image processing technology in comprehensive grape plant research.

The new study emphasizes the efficacy of combining advanced preprocessing steps with machine learning to achieve superior accuracy in grape variety classification, reinforcing previous research findings while providing a focused contribution to color-based classification methods. The comparison of the results of this research with the results of previous studies can be seen in Table 7. Previous Study 1 used KNN and SVM to classify grape leaves, achieving 96.66% accuracy with KNN and 90% with SVM. This research outperformed with 98.89% accuracy using a 70:30 data split, indicating that more advanced image processing methods can yield better results. Previous Study 2 employed KNN for grape image classification based on texture features, achieving 99.5441% accuracy. Although this study achieved slightly higher accuracy, it focused on texture features. This research's focus on color features also demonstrates high effectiveness with 98.89% accuracy, underscoring the potential of color-based methods. Previous Study 3 focused on grape berry detection and size measurement using edge image processing. This research aligns with the use of image processing in grape research. Still, it adds value by providing a method for classifying grape varieties based on color, offering practical benefits for agricultural monitoring and management.

Table 7. Comparison Of New Research And Previous Studies

Study	Method	Focus	Accuracy	Comparison
Study 1	KNN, SVM	Grape leaves classification	KNN: 96,66%, SVM: 90%	This research achieved higher accuracy (98.89%) using more complex methods.
Study 2	KNN	Grape image classification based on texture	99,5441%	There is slightly higher accuracy, but this research focuses on highly effective color features (98.89%).
Study 3	Edge image processing	Grape berry detection and size measurement	Not specified	Supports image processing in grape research; this research adds classification based on color for practical applications.

g. Application Implementation The following are screenshots of the GUI interface for classifying grape varieties based on color feature extraction using the Naïve Bayes method. The classification results for each type of grapegreen, black, and redare shown in Figure 8, Figures 9, and Figure 10.

Figure 8. GUI Interface for Green Grape Classification

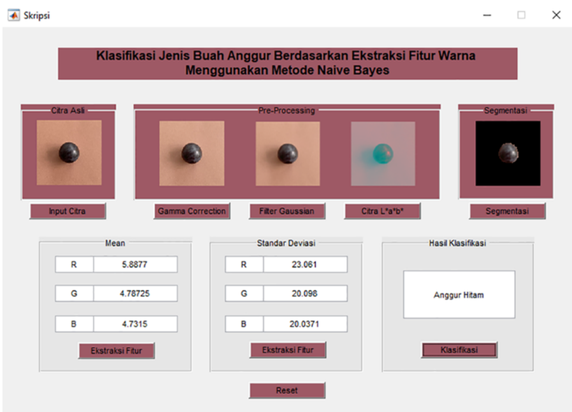


Figure 9. GUI Interface for Black Grape Classification

Figure 10. Interface for Red Grape Classification

h. Discussion

This research presents a comprehensive approach to grape variety classification using image processing techniques. The high accuracy achieved in both within-database and out-of-database testing scenarios underscores the robustness of the methodology. The combination of gamma correction, Gaussian filtering, color space transformation, K-Means clustering, and statistical feature extraction proved to be highly effective. The findings contribute to the field by providing a reliable method for grape classification, which can benefit agricultural practices and research. Future work could explore integrating texture features and more advanced machine learning algorithms to enhance the system's performance further.

4. CONCLUSION

This study successfully developed a grape classification system using the Naïve Bayes method based on color feature extraction. The primary objective was to classify grape varieties using image processing techniques. The novel contribution of this research lies in integrating meticulous data preprocessing steps, including gamma correction, Gaussian filtering, and color conversion, which significantly enhance the accuracy of the classification system. The system achieved an impressive accuracy of 98.89% with a 70:30 data split. This method's novelty is highlighted by its high performance in grape classification, demonstrating the effectiveness of focusing on color features. While the study is limited by its reliance on color characteristics and a relatively small dataset, it paves the way for future research to integrate additional features and expand the dataset. The developed methodology holds significant potential for practical applications in agricultural settings, particularly for grape plant monitoring and management, thus contributing to the advancement of sustainable farming practices.

5. ACKNOWLEDGEMENTS

The Acknowledgments section is optional. Research sources can be included in this section.

6. DECLARATIONS

AUTHOR CONTIBUTION

Putri Jafar: Conducted analysis, conceived and designed the research, and collected image data. Dolly Indra: Served as the main supervisor and was responsible for conducting reviews. Fitriyani Umar: Served as the second supervisor, responsible for conducting reviews.

FUNDING STATEMENT

Self-funded by the author.

COMPETING INTEREST

The authors declare that there are no conflicts of interest related to the publication of this article.

REFERENCES

[1] J. P. O. Li, H. Liu, D. S. Ting, S. Jeon, R. V. Chan, J. E. Kim, D. A. Sim, P. B. Thomas, H. Lin, Y. Chen, T. Sakomoto, A. Loewenstein, D. S. Lam, L. R. Pasquale, T. Y. Wong, L. A. Lam, and D. S. Ting, "Digital technology, tele-medicine and artificial intelligence in ophthalmology: A global perspective," *Progress in Retinal and Eye Research*, vol. 82, no. June 2020, 2021, https://doi.org/10.1016/j.preteyeres.2020.100900.

[2] A. Alam, "Employing Adaptive Learning and Intelligent Tutoring Robots for Virtual Classrooms and Smart Campuses: Reforming Education in the Age of Artificial Intelligence," in *Lecture Notes in Electrical Engineering*, R. N. Shaw, S. Das, V. Piuri, and M. Bianchini, Eds., vol. 914. Singapore: Springer Nature Singapore, 2022, pp. 395–406, https://doi.org/10.1007/978-981-19-2980-9_32.

[3] A. A. Bharate and M. S. Shirdhonkar, "Classification of Grape Leaves using KNN and SVM Classifiers," *Proceedings of the 4th International Conference on Computing Methodologies and Communication, ICCMC 2020*, no. Iccmc, pp. 745–749, 2020, https://doi.org/10.1109/ICCMC48092.2020.ICCMC-000139.

[4] Pulung Nurtantio Andono and S. H. Nugraini, "Texture Feature Extraction in Grape Image Classification Using K-Nearest Neighbor," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 5, pp. 768–775, 2022, https://doi.org/10.29207/resti.v6i5.4137.

[5] L. Luo, W. Liu, Q. Lu, J. Wang, W. Wen, D. Yan, and Y. Tang, "Grape berry detection and size measurement based on edge image processing and geometric morphology," *Machines*, vol. 9, no. 10, 2021, https://doi.org/10.3390/machines9100233.

[6] R. Malani, A. B. W. Putra, and M. Rifani, "Implementation of the naive bayes classifier method for potential network port selection," *International Journal of Computer Network and Information Security*, vol. 12, no. 2, pp. 32–40, 2020, https://doi.org/10.5815/ijcnis.2020.02.04.

[7] M. R. Romadhon and F. Kurniawan, "A Comparison of Naive Bayes Methods, Logistic Regression and KNN for Predicting Healing of Covid-19 Patients in Indonesia," *3rd 2021 East Indonesia Conference on Computer and Information Technology, EIConCIT 2021*, pp. 41–44, 2021, https://doi.org/10.1109/EIConCIT50028.2021.9431845.

[8] S. Singh, N. K. Garg, and M. Kumar, "Feature extraction and classification techniques for handwritten Devanagari text recognition: a survey," *Multimedia Tools and Applications*, vol. 82, no. 1, pp. 747–775, 2023, https://doi.org/10.1007/s11042-022-13318-9.

[9] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," *Global Transitions Proceedings*, vol. 3, no. 1, pp. 91–99, 2022, https://doi.org/10.1016/j.gltp.2022.04.020.

[10] T. Rahman, A. Khandakar, Y. Qiblawey, A. Tahir, S. Kiranyaz, S. B. Abul Kashem, M. T. Islam, S. Al Maadeed, S. M. Zughaier, M. S. Khan, and M. E. Chowdhury, "Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images," *Computers in Biology and Medicine*, vol. 132, no. November 2020, 2021, https://doi.org/10.1016/j.compbiomed.2021.104319.

[11] A. Kumar and S. S. Sodhi, "Comparative analysis of gaussian filter, median filter and denoise autoenocoder," *Proceedings of the 7th International Conference on Computing for Sustainable Global Development, INDIACom 2020*, vol. 6, pp. 45–51, 2020, https://doi.org/10.23919/INDIACom49435.2020.9083712.

[12] C. Li, S. Anwar, J. Hou, R. Cong, C. Guo, and W. Ren, "Underwater Image Enhancement via Medium Transmission-Guided Multi-Color Space Embedding," *IEEE Transactions on Image Processing*, vol. 30, pp. 4985–5000, 2021, https://doi.org/10.1109/TIP.2021.3076367.

[13] S. Jadon, "A survey of loss functions for semantic segmentation," *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology, CIBCB 2020*, 2020, https://doi.org/10.1109/CIBCB48159.2020.9277638.

[14] W. Xiaoqiong and Y. E. Zhang, "Image segmentation algorithm based on dynamic particle swarm optimization and K-means clustering," *International Journal of Computers and Applications*, vol. 42, no. 7, pp. 649–654, 2020, https://doi.org/10.1080/1206212X.2018.1521090.

[15] Y. E. Yana, T. Informatika, F. Teknik, and U. I. Lamongan, "Klasifikasi Jenis Pisang Berdasarkan Fitur Warna , Tekstur , Bentuk Citra Menggunakan SVM dan KNN," vol. 4, no. 1, pp. 28–36, 2021, https://doi.org/10.25273/research.v4i1.6687.

[16] S. Rani, K. Lakhwani, and S. Kumar, *Three dimensional objects recognition & pattern recognition technique; related challenges: A review*. Multimedia Tools and Applications, 2022, vol. 81, no. 12, https://doi.org/10.1007/s11042-022-12412-2.

[17] J. G. Perez and E. S. Perez, "Predicting Student Program Completion Using Naïve Bayes Classification Algorithm," *International Journal of Modern Education and Computer Science*, vol. 13, no. 3, pp. 57–67, 2021, https://doi.org/10.5815/IJMECS.2021.03.05.

[18] D. Syafira, S. Suwilo, and P. Sihombing, "Analysis of classification and Naïve Bayes algorithm k-nearest neighbor in data mining," *IOP Conf. Series: Materials Science and Engineering*, 2020, https://doi.org/10.1088/1757-899X/725/1/012106.