Journal of Applied Informatics and Computing (JAIC)
Vol.9, No.4, August 2025, pp. 1793~1799
e-ISSN: 2548-6861

Image Classification of Red Dragon Fruit Ripeness Levels Using HSV Color Moments and the K-NN Algorithm

Nadia Br Sembiring [1]*, M. Fakhriza [2]
Ilmu Komputer, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sumatera Utara, Medan, Indonesia
nadiaintansaqinah@gmail.com[1], fakhriza@uinsu.ac.id[2]

Article Info

Article history:

Received 2025-07-13
Revised 2025-07-21
Accepted 2025-07-30

Keyword:

Red Dragon Fruit,
HSV,
Color Moment,
K-Nearest Neighbor,
Digital Image Processing.

ABSTRACT

Accurately determining the ripeness level of red dragon fruit (Hylocereus polyrhizus) is crucial for ensuring post-harvest quality and distribution efficiency. This study proposes a method for classifying red dragon fruit ripeness using color moment features in the HSV color space combined with the K-Nearest Neighbor (K-NN) algorithm. The dataset consists of 2,881 images of dragon fruit with a resolution of 800×800 pixels, categorized into three classes: ripe (886 images), unripe (1,241 images), and rotten (754 images). All images were captured under natural lighting conditions and underwent pre-processing to enhance color value consistency. Color features were extracted by calculating the mean, standard deviation, and skewness of the Hue, Saturation, and Value channels. The K-NN model was trained and tested on data randomly split in an 80:20 ratio. The testing results showed that the model achieved 100% accuracy in classifying the ripeness levels, demonstrating the effectiveness of this non-destructive method in distinguishing fruit ripeness. This approach holds strong potential to support efficient and consistent decision-making in the agricultural sector.

I. Introduction

Red dragon fruit (*Hylocereus polyrhizus*) is a horticultural commodity with high economic value and significant health benefits. Its high vitamin C, antioxidant, and fiber content makes this fruit popular in both domestic and international markets. Market demand for dragon fruit continues to increase, along with increasing public awareness of a healthy lifestyle and promising export potential. Therefore, post-harvest handling, particularly in determining the level of fruit ripeness, is a crucial factor in maintaining product quality and competitiveness [1].

Manually determining the ripeness level of dragon fruit relies solely on visual observation of skin color and the subjective experience of farmers. However, this method has drawbacks, such as inconsistent results and reliance on the subjectivity of sorting operators, which in certain situations can result in inconsistent classification processes [2]. Errors in determining ripeness level can have serious consequences, ranging from overripe fruit that is prone to rotting to under ripe fruit that is less desirable to consumers. Furthermore, this manual method is quite time-consuming and inefficient when applied on a large production scale. Thus, a more accurate and efficient alternative method is needed to determine the ripeness level of dragon fruit.

As an alternative, various technology-based non-destructive approaches have been developed to automatically identify fruit ripeness. One promising approach is the use of digital images and artificial intelligence-based classification algorithms [5]. Previous research has shown that these algorithms *K-Nearest Neighbor* (KNN) is effective in visual object classification tasks in agriculture, including fruit ripeness classification based on skin color [3]. For example, research by Hernando et al. (2024) showed that the use of KNN was able to achieve accuracy of up to 91% in classifying dragon fruit ripeness based on skin color, while Munandar (2022) reported accuracy of up to 99.25% using a color histogram approach [4].

In the context of color-based image classification, the choice of color space plays a crucial role in determining the quality of extracted features. This study employs the HSV (Hue, Saturation, Value) color space due to its ability to separate pure color information (hue) from light intensity (value), allowing the color features to be more stable under varying natural lighting conditions typically encountered during field image acquisition. Compared to the RGB color space, which is sensitive to lighting changes, HSV better reflects human visual perception of color. While the LAB color space also offers perceptual uniformity, it is more computationally complex and less commonly used in simple color moment-based classification [7]. Therefore, this study aims to develop a classification model for red dragon fruit ripeness using a combination of HSV Color Moment features and the KNN algorithm on a well-structured dataset. With this approach, it is hoped that a non-destructive classification system can be obtained that is efficient and accurate and can support more objective decision-making in the agricultural sector [6].

II. METHOD

A. *Data* Collection Techniques

The following are the data collection techniques used to support this research:

1) Observation, Data was obtained directly from the object of research, namely dragon fruit. This data collection process is carried out by taking pictures of dragon fruits directly in the garden or cultivation location using cameras or special devices. The images taken include dragon fruits in various levels of ripeness, ranging from raw to ripe [9].
2) Literature Studies, Literature studies are obtained from various literatures, scientific journals, previous research, and other reliable sources that discuss the Classification of the ripeness level of dragon fruit, image processing methods, and the application of the K-Nearest Neighbour (K-NN) algorithm. This secondary data serves as a supporting reference in compiling a research framework, designing methods, and comparing research results with previous studies [8][10].

B. *Data Analysis*

In the data analysis stage, this study utilizes digital images as the primary data source for recognizing the ripeness level of dragon fruit. The dataset used consists of a total of 2,881 images of red dragon fruit, with each image having a resolution of 800×800 pixels. These images were acquired under consistent natural lighting conditions and fixed perspectives to ensure uniformity across samples. The dataset is divided into three ripeness categories: unripe (immature) with 1,241 images, ripe (mature) with 886 images, and overripe or defective (rotten) with 754 images. RGB (Red, Green, Blue) images were initially used due to their capacity to represent color variations in detail, which is crucial for visual analysis in detecting differences in skin color that correspond to varying ripeness stages. The color of the dragon fruit skin, which changes noticeably throughout the ripening process, serves as the main visual indicator in this classification task.

The collected image data are then processed and analyzed using feature extraction techniques combined with a machine learning approach, specifically the K-Nearest Neighbour (K-NN) algorithm. This analysis focuses on identifying visual patterns related to the ripening process, such as hue, saturation, and brightness variations on the fruit's surface. The accuracy of the resulting classification model is strongly influenced by the quality and quantity of the input image data, as well as the balance of class distribution across different ripeness levels during the training and testing phases.

C. *System Planning*

The system design in the form of a flowchart can be seen as follows.

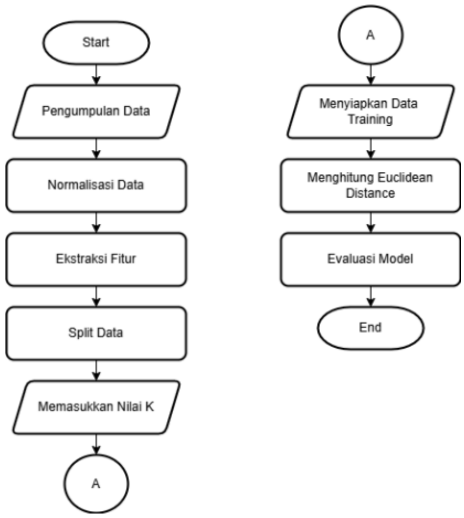

Figure 1. Flowchart System

Figure 2 illustrates the recognition process flow, which begins with the data collection stage, where the system gathers images or color data of dragon fruit to be used for recognition purposes. Once the data is collected, a normalization process is carried out to ensure that all data are on a uniform scale, thereby reducing the risk of bias due to differences in value ranges. Next, the system performs feature extraction by retrieving color information from the dragon fruit such as RGB or HSV values which serves as the basis for determining the ripeness level. The extracted data is then divided into training and testing sets, which are used to train and evaluate the performance of the KNN model. Before the recognition process begins, the user must define the value of K, representing the number of nearest neighbors used as a reference in determining the ripeness class of the dragon fruit.

Once the K value is defined, the system proceeds to prepare the training data, where the preprocessed training data is used as a reference for classification. The system then calculates the Euclidean distance between each test sample and all available training samples. This distance calculation is aimed at measuring the similarity between test data and training data based on the extracted color features. Training samples with the shortest distance are considered the nearest neighbors and are used as references in determining the ripeness class of the test sample. Following this, the model undergoes evaluation to assess how accurately the system classifies the ripeness level of the dragon fruit. This evaluation allows for potential improvements to the model, such as adjusting the K value or adding more training data. Finally, the classification process is completed, and the model is ready to be used for automatically identifying the ripeness level of dragon fruit based on its color.

D. Classification

The selection of the K parameter in the K-Nearest Neighbour (KNN) algorithm was carried out through experimentation by testing several odd values of K, namely 1, 3, 5, 7, and 9. The performance evaluation was conducted on a subset of the test data by observing the accuracy metric. Based on the experimental results, the value of **K = 3** yielded the highest and most consistent accuracy, and was therefore chosen as the final parameter in the model implementation. This selection aimed to achieve a balance between model complexity and its ability to generalize to new data.

E. Implementation

In the implementation stage, a system for recognizing the level of ripeness of dragon fruit based on color was built using the C# programming language and implemented on the platform. The implementation process begins with processing dragon fruit color data obtained from images uploaded by users. The data is then extracted using the HSV Color Moment method to obtain representative features used in the classification process. The K-Nearest Neighbor (K-NN) method is applied by calculating the Euclidean distance between the test data and the training data based on the resulting feature values, in order to determine the maturity level based on the number of nearest neighbors (K) that has been previously determined. The recognition results are then displayed to the user through the application interface. Which is designed to be easy to use and provides visual information regarding the ripeness status of dragon fruit.

F. Testing

To produce an optimal dragon fruit ripeness classification system based on color, researchers need to test the system that has been developed. In the context of this research, researchers will test the classification system that has been implemented on a desktop platform using the C# programming language. Testing is carried out using a dataset of dragon fruit images at various levels of ripeness that have been labeled as training data and test data. Next, several test scenarios will be developed that aim to measure the accuracy of the classification model with the K-Nearest Neighbor (K-NN) algorithm in classifying the level of fruit ripeness based on color features extracted using the HSV method. *Color Moment* The goal of this research is to achieve optimal accuracy in dragon fruit ripeness classification, with an expected accuracy ranging from 80% to 100%. Therefore, the K-NN method implemented in this desktop application is expected to be effective and accurate in assisting in the automatic identification of dragon fruit ripeness.

III. RESULT AND DISCUSSION

A. Research Data

The data used in this study is a public image dataset entitled. The use of this standard dataset aims to ensure the validity and reproducibility of research results. This dataset contains a total of 2,881 images of red dragon fruit that have been classified into three different classes of maturity levels. The distribution of the original data from the imagery for each class can be seen in table 1.

TABLE 1
DATA DISTRIBUTION

No	Class	Category	Number of Images
1	Mature	Mature	886
2	Immature	Immature (Raw)	1241
3	Defect	Poor Maturity (Rotten)	754

Each image in the dataset has a resolution of 800x800 pixels in JPG file format. This clearly structured and labeled dataset is a solid basis for the training and testing phase of the proposed K-Nearest Neighbor (KNN) classification model. A picture of the dragon fruit dataset can be seen in figure 3.

Figure 2. Ripe Dragon Fruit (Left), Raw (Middle), Rotten (Right)

B. Application of the K-Nearest Neighbour (KNN) Method

The developed K-Nearest Neighbour (KNN) model successfully classified red dragon fruit images based on their ripeness into three categories: ripe, unripe, and rotten. The dataset consisted of 2,881 images, and 500 images were randomly selected as a subset for training and testing, with a ratio of 80:20. The model was trained using 400 data sets and tested using 100 data sets that had never been involved in previous training [11].

The test results show that the model is capable of classifying with 100% accuracy, as indicated by the confusion matrix. Each test image was correctly classified into its original class. There were no misclassifications (false positives or false negatives), resulting in a maximum accuracy metric. This indicates that the HSV Color Moment statistical feature effectively represents the visual differences between dragon fruit ripeness classes [12].

C. Data Input and RGB Image Representation

The process begins with a digital image in RGB format. Let's assume a 5x5 pixel area of the image is represented by the following matrix. Each cell contains a value (Red, Green, Blue), where the center area represents the dragon fruit object (reddish) and the edge area represents the background (greenish).

TABLE 2
RGB MATRIX REPRESENTATION OF A 5X5 PIXEL IMAGE

(40,150,30)	(55,165,40)	(60,170,55)	(50,160,35)	(45,155,33)
(50,160,38)	(210,50,90)	(220,60,100)	(215,55,95)	(58,168,45)
(60,170,50)	(205,45,85)	(225,65,105)	(218,60,98)	(62,172,58)
(55,165,42)	(208,48,88)	(212,52,92)	(222,62,102)	(53,163,41)
(48,158,36)	(65,175,60)	(70,180,65)	(68,178,63)	(52,162,39)

D. Segmentation with Thresholding

The next step is to separate the Region of Interest (ROI) from the background. A simple thresholding method is used on the Red channel. Pixels will be retained if their R value is greater than the threshold R > 200. Pixels that do not meet the requirements have their values set to (0,0,0).

TABLE 3
RGB MATRIX SEGMENTATION RESULTS

(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)
(0,0,0)	(210,50,90)	(220,60,100)	(215,55,95)	(0,0,0)
(0,0,0)	(205,45,85)	(225,65,105)	(218,60,98)	(0,0,0)
(0,0,0)	(208,48,88)	(212,52,92)	(222,62,102)	(0,0,0)
(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)

1) RGB to HSV Conversion

Next, the initial pixels will be calculated which have values based on table 4.3, namely RGB = (210,50,90)

Step 1: Normalize RGB Values
Change the value from the range [0-255] to the range [0-1].
R = 210 / 255 = 0.82
G = 50 / 255 = 0.20
B = 90 / 255 = 0.35

Step 2: Find the Maximum (Cmax) and Minimum (Cmin) Values
Cmax = max(R, G, B) = 0.82 (from R)
Cmin = min(R, G, B) = 0.20 (dari G)

Step 3: Calculate Value (V)
Value is Cmax.
V = 0.82

Step 4: Calculate Saturation (S)
Formula: S = (Cmax - Cmin) / Cmax
S = (0.82 - 0.20) / 0.82
S = 0.62 / 0.82
S ≈ 0.76
Step 5: Calculate Hue (H)
The Hue formula depends on which component is Cmax. Since Cmax is R', the formula for the R channel is:
H' = 60° * ((G' - B') / (Cmax - Cmin)) mod 6
H' = 60° * ((0.20 - 0.35) / (0.82 - 0.20))
H' = 60° * (-0.15 / 0.62)
H' = 60° * (-0.24) = -14.4°

Since the result is negative, then add 360°:
H = -14.4° + 360° = 345.6°

So the RGB pixel values (210, 50, 90) after being converted to HSV ≈ (H=345.6°, S=0.76, V=0.82). Repeat the calculation until all pixels have been converted to HSV. The matrix of the HSV conversion calculation results can be seen in table 4.

TABLE 4
HSV CONVERSION MATRIX

(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)
(0,0,0)	(345.6, 0.76, 0.82)	(345.0, 0.73, 0.86)	(345.0, 0.74, 0.84)	(0,0,0)
(0,0,0)	(345.0, 0.78, 0.80)	(345.0, 0.71, 0.88)	(345.0, 0.73, 0.85)	(0,0,0)
(0,0,0)	(345.0, 0.77, 0.82)	(345.0, 0.75, 0.83)	(345.0, 0.72, 0.87)	(0,0,0)
(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)

Mathematically, the RGB to HSV conversion formula produces a Hue (H) value in degrees, which is in the range [0-360]. So that the Hue value fits in a more efficient data type, it needs to be scaled. The purpose of this scaling is so that the Hue value can be stored in one *byte* (8-bit) with a value range of 0-255, so that the Hue value will be divided by 2. For example at pixel (345.6, 0.76, 0.82), the value 345.6 / 2 = 172.8 so that if rounded it becomes 173. Based on the calculation, then the matrix of the scaling results of the Hue value can be seen in table 5.

TABLE 5
HSV CONVERSION RESULT MATRIX

(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)
(0,0,0)	(173, 0.76, 0.82)	(172, 0.73, 0.86)	(172, 0.74, 0.84)	(0,0,0)
(0,0,0)	(172, 0.78, 0.80)	(172, 0.71, 0.88)	(172, 0.73, 0.85)	(0,0,0)
(0,0,0)	(172, 0.77, 0.82)	(172, 0.75, 0.83)	(172, 0.72, 0.87)	(0,0,0)
(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)

Next, based on the HSV conversion matrix, the feature extraction value will be calculated using *color moment* which will be discussed in the next section.

2) Calculation Color Moment

Feature calculation *Color Moment* performed on the remaining 9 pixels in the ROI (N=9). Based on the segmentation result matrix, the first three statistical moments are calculated using the following formula.

a. Mean (No)

Mean represents the average color value in each channel [13] [14] [15].

$$AND_i = \sum_{N}^{j=1} \frac{1}{N} P_{ij} \quad (1)$$

b. Standard Deviation (σi)

Standard deviation measures the spread or variation of color from its mean [13].

$$\sigma_i = \sqrt{\left(\frac{1}{N} \sum_{N}^{j=1} \left(P_{ij} - AND_i\right)^2\right)} \quad (2)$$

c. Skewness (*And*)

Skewness measures the degree of asymmetry of the color distribution [13].

$$S_i = \sqrt[3]{\left(\frac{1}{N} \sum_{N}^{j=1} \left(P_{ij} - AND_i\right)^2\right)} \quad (3)$$

Up to value *skewnes* of the HSV channel is (H=0.43, S=0.02, V=0.01). After completing all calculations, 9 feature values are obtained.*color moment* from the 5x5 image as follows.

a. Mean (E): (H=172.11, S=0.74, V=0.84)
b. Standard Deviation (σ): (H=0.31, S=0.02, V=0.02)
c. Skewness (S): (H=0.43, S=0.02, V=0.01)

Thus, the 9-dimensional feature vector for the image in table 2 is

Features = [172.11, 0.74, 0.84, 0.31, 0.02, 0.02, 0.43, 0.02, 0.01]

TABLE 6
FEATURE VECTORS OF 5X5 IMAGE DATA

Index	0	1	2	3	4	5	6	7	8
Value	172.11	0.74	0.84	0.31	0.02	0.02	0.43	0.02	0.01

E. Splitting Dataset

From the total dataset of 2881 data, a subset of 500 randomly selected data will be used to train and test the model.

This dataset is then divided using a ratio of 80% for the training data and 20% for the test data:

1. Training Data: Consists of 400 pieces of data, used so that the model can "learn" the characteristics of the features of each class.
2. Testing Data: Consists of 100 pieces of data, used to evaluate the model's performance on never-before-seen data.

F. Training Data

KNN is known as the lazy learning algorithm. This term is used because the training process does not involve the construction of complex mathematical models like other algorithms. On the other hand, the training process in KNN is only in the form of storing all feature vectors from 400 training data and their class labels ('Mature', 'Raw', 'Rotten') into memory.

To provide a visual overview of a single data stored, Table 2 presents an example of a single feature vector of the training data for the 'Mature' class, based on the results of our previous manual calculations.

TABLE 7
A SINGLE VECTOR REPRESENTATION OF TRAIN DATA FEATURE

Feature Index	Description	Value
1	Mean Hue	172.11
2	Mean Saturation	0.74
3	Mean Value	0.84
4	Standar Deviasi Hue	0.31
5	Standar Deviasi Saturation	0.02
6	Standar Deviasi Value	0.02
7	Skewness Hue	0.43
8	Skewness Saturation	0.02
9	Skewness Value	0.01

G. Testing Data

To demonstrate the KNN testing workflow in detail, this subsection presents an example of manual classification calculations. We will use 10 hypothetical training datasets divided into two classes ('Ripe' and 'Raw'), and one test dataset for which we will determine the class.

1) Training Data

Assumed we have 10 feature vectors in the training data.

a. 5 'Mature' class data.
b. 5 'Raw' class data.

The first data is the result of calculations that have been done previously. The other data is the data*dummy*which aims to calculate distance and process*voting* in the KNN algorithm. The training dataset table of 10 can be seen in the following table.

TABLE 8
HYPOTHETICAL TRAINING DATASET

ID	Class	mH	mS	mV	StdH	StdS	StdV	SkwH	SkwS	SkwV
D1	KIND	172.11	0.74	0.84	0.31	0.02	0.02	0.43	0.02	0.01
D2	KIND	170.5	0.78	0.88	0.40	0.03	0.03	0.35	0.01	0.02
D3	KIND	173.2	0.71	0.81	0.35	0.02	0.01	0.48	0.03	0.01
D4	KIND	169.8	0.80	0.90	0.28	0.01	0.02	0.30	0.01	0.01
D5	KIN	171.5	0.75	0.85	0.33	0.02	0.02	0.40	0.02	0.02

	D									
D6	Raw	80.2	0.65	0.70	2.50	0.08	0.05	0.80	0.05	0.04
D7	Raw	85.5	0.61	0.68	2.80	0.09	0.06	0.75	0.06	0.05
D8	Raw	78.9	0.70	0.72	2.65	0.07	0.04	0.82	0.04	0.03
D9	Raw	82.1	0.63	0.69	2.70	0.08	0.05	0.78	0.05	0.04
D10	Raw	81.3	0.68	0.71	2.55	0.07	0.04	0.79	0.04	0.03

2) Data Test

In the test stage, a hypothetical test feature vector is used. These vectors will be classified by the model to determine their class predictions.

TABLE 9
VECTOR TESTING FEATURES

mH	mS	mV	StdH	StdS	StdV	SkwH	SkwS	SkwV
171.8	0.76	0.86	0.32	0.02	0.02	0.41	0.02	0.01

H. Classification Process KNN (K=3)

The next stage is to classify the testing vector data using the KNN algorithm assuming the selected parameters K=3, based on 10 training data that has been prepared. The classification stages with KNN are as follows.

1) Euclidean Distance Calculation, Distance calculation is carried out 10 times a number of training data.

TABLE 10
RESULTS OF EUCLIDEAN DISTANCE CALCULATION

ID	Kelas	Jarak
D1	Mature	0.302
D2	Mature	1.386
D3	Mature	1.487
D4	Mature	2.088
D5	Mature	0.283
D6	Raw	91.68
D7	Raw	86.38
D8	Raw	92.98
D9	Raw	89.78
D10	Raw	90.58

2) Sort Distances and find Nearest Neighbor K

TABEL 11
DATA SEQUENCING RESULTS BY DISTANCE

ID	Class	Distance
D5	**Mature**	**0.283**
D1	**Mature**	**0.302**
D2	**Mature**	**1.386**
D3	Mature	1.487
D4	Mature	2.088
D6	Raw	91.68
D7	Raw	86.38
D8	Raw	92.98
D9	Raw	89.78
D10	Raw	90.58

3) Based on the parameter K = 3, the three closest neighbors are:
   a. D5 Class: Mature
   b. D1 Class: Mature
   c. D2 Class: Mature

The voting results obtained 3 votes for the "Matang" class, and 0 votes for the "Raw" class.

I. Model Testing and Accuracy

The high accuracy indicates that the KNN algorithm is highly suitable for fruit ripeness classification based on color features.

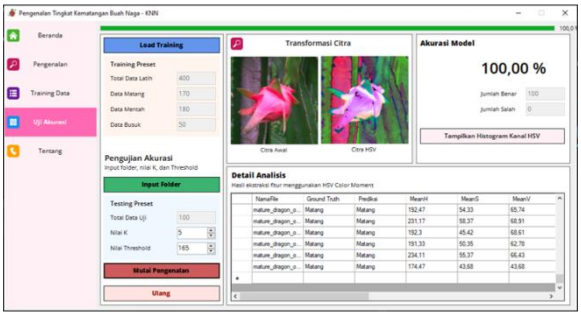

Figure 3. Confusion Matrix Testing View

The advantages of this method lie in its simplicity and its ability to recognize patterns based on the proximity between features. This optimal performance is also influenced by the quality of the well-labeled dataset and image pre-processing that minimizes noise caused by lighting and background.

TABLE 12
CONFUSION MATRIX TESTING

		Prediction		
		Mature	Raw	Rotten
Actual	Mature	40	0	0
	Raw	0	40	0
	Rotten	0	0	20

Based on the values from the Confusion Matrix above, the model's performance is measured using the Accuracy metric. Accuracy measures how much of the data was successfully correctly predicted from the overall test data.

$$Akurasi = \frac{Jumlah\ Prediksi\ Benar}{Total\ Seluruh\ Data\ Uji} \times 100\%$$

$$Akurasi = \frac{100}{100} \times 100\% = 100\%$$

From the calculation results, it was obtained that the model accuracy level was 100% in distinguishing the ripeness level of dragon fruit. This accuracy was obtained from testing 100 randomly selected test data, where the images on the test data were never used during the model training process.

J. System Interface

As a form of implementation of the research, the entire As a practical implementation of the proposed method, the entire classification process has been implemented in a desktop-based system. This system was built to automatically process dragon fruit images, extract HSV Color Moment features, and apply classification using the K-Nearest Neighbor (KNN) algorithm. The system's functionality includes training data management, test image processing, and model performance evaluation through accuracy calculations based on the input test data.

The system's interface is designed to be simple and functional to facilitate user access to key features, such as dataset loading, image processing, and classification result display. This system not only simplifies dragon fruit ripeness classification but also provides direct visualization of results and evaluation of model performance, making it a useful decision-making tool in the field.

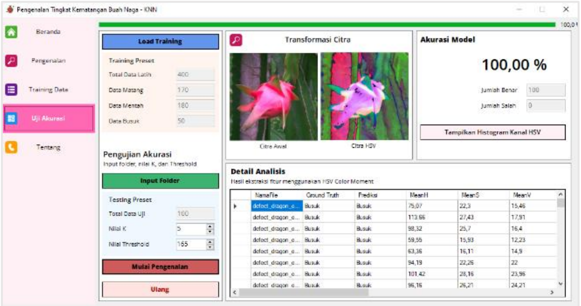

Figure 4. Accuracy Test View

IV. CONCLUSION

This research has implemented the proposed method into a desktop application system. This system is capable of automatically classifying the ripeness level of red dragon fruit based on 800 x 800 pixel digital images. The implemented method is K-Nearest Neighbor (KNN) which works based on nine features extracted using HSV Color Moment, which includes the Mean, Standard Deviation, and Skewness of each color channel.

In testing conducted on 100 test data sets out of a total of 500 data sets used, with 400 data sets as training data, the proposed model demonstrated very high performance by achieving an accuracy rate of 100%. Based on these test results, it can be concluded that the HSV Color Moment features are very effective and are able to clearly differentiate between maturity classes. These results also indicate that the KNN algorithm, despite its simplicity, has excellent capabilities in mapping these features into the correct classification of this study dataset.

BIBLIOGRAPHY

[1] A. C. Silor, F. S. C. Silor, and M. B. Emfimo, “Exploring the Economic and Health Benefits of Dragon Fruit : Consumer Perceptions , Market Dynamics , and Strategic Development for Filipino Farmers,” vol. XI, no. 2321, pp. 420–432, 2024, doi: 10.51244/IJRSI.

[2] M. F. Barkah, “Klasifikasi Rasa Buah Jeruk Pontianak Berdasarkan Warna Kulit Buah Jeruk Menggunakan Metode K-Nearest Neighbor,” *Coding J. Komput. dan Apl.*, vol. 8, no. 1, 2020, doi: 10.26418/coding.v8i1.39193.

[3] Syafa’ati, D. M. R. Budiastra, and I. Wayan, “Penentuan Tingkat Kekerasan dan Kemanisan Buah Naga Merah (Hylocereus polyrhizus) Secara Nondestruktif Menggunakan Near Infrared Spectroscopy (NIRS),” 2021.

[4] E. Hernando, A. C. Saputra, and J. Parhusip, “Klasifikasi Kematangan Buah Naga Berdasarkan Warna Kulit Menggunakan Algoritma K-Nearest Neighbor,” *AnoaTIK*, 2024.

[5] W. R. U. Fadilah, D. Agfiannisa, and Y. Azhar, “Analisis Prediksi Harga Saham PT. Telekomunikasi Indonesia Menggunakan Metode Support Vector Machine,” *Fountain Informatics J.*, vol. 5, no. 2, p. 45, 2020, doi: 10.21111/fij.v5i2.4449.

[6] K. Martin, K. Shilton, and J. Smith, “Business and the ethical implications of technology: Introduction to the symposium,” *Bus. Ethical Implic. Technol.*, pp. 1–11, 2022, doi: 10.1007/s10551-019-04213-9.

[7] J. A. Munandar, “Deteksi Kematangan Buah Naga Menggunakan Fitur Histogram Warna Berbasis Computer Vision,” 2022.

[8] N. Alkarim and R. A. Zuama, “Implementasi Algoritma Knn (K-Nears Neighbor) Pada Prediksi Harga Mobil Bekas,” *J. IJCCS*, vol. 3, no. 1, pp. 1–7, 2024.

[9] A. Khair, V. Rosalina, and S. S, “Rancang Bangun Sistem Informasi E-Commerce Dengan Penerapan Customer Relationship Management Berbasis Web,” *PROSISKO J. Pengemb. Ris. dan Obs. Sist. Komput.*, vol. 8, no. 2, pp. 60–85, 2021, doi: 10.30656/prosisko.v8i2.3856.

[10] G. F. Grandis, Y. Arumsari, and Indriati, “Seleksi Fitur Gain Ratio pada Analisis Sentimen Kebijakan Pemerintah Mengenai Pembelajaran Jarak Jauh dengan K-Nearest Neighbor,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 8, pp. 3507–3514, 2021.

[11] Z. Mahendra and A. Ridok, “Analisis Sentimen Opini Masyarakat Terhadap Fenomena TikTokShop di Indonesia Menggunakan Metode K-Nearest Neighbor berbasis N-gram dengan Seleksi Fitur Information Gain,” vol. 1, no. 1, pp. 1–10, 2017.

[12] A. Halim, Y. Yusra, M. Fikry, M. Irsyad, and E. Budianita, “Klasifikasi Sentimen Masyarakat Di Twitter Terhadap Prabowo Subianto Sebagai Bakal Calon Presiden 2024 Menggunakan M-KNN,” *J. Inf. Syst. Res.*, vol. 5, no. 1, pp. 202–212, 2023, doi: 10.47065/josh.v5i1.4054.

[13] B. Maulana Alfaruq, D. Erwanto, and I. Yanuartanti, “Klasifikasi Kematangan Buah Tomat Dengan Metode Support Vector Machine,” *Gener. J.*, vol. 7, no. 3, pp. 64–72, 2023, doi: 10.29407/gj.v7i3.21092.

[14] D. E. Kurniawan, N. R. Hartadi, and P. Prasetyawan, "Analisis Hasil Teknik Penyembunyian Hak Cipta Menggunakan Transformasi DCT dan RSPPMC pada Jejaring Sosial," Jurnal Teknologi Informasi dan Ilmu Komputer, vol. 5, no. 3, Art. no. 3, Aug. 2018, doi: 10.25126/jtiik.201853692.

[15] R. I. Borman, D. E. Kurniawan, Styawati, I. Ahmad, and D. Alita, “Classification of maturity levels of palm fresh fruit bunches using the linear discriminant analysis algorithm,” AIP Conf. Proc., vol. 2665, no. 1, pp. 30023.1–30023.8, 2023, doi: 10.1063/5.0126513..