BAREKENG: Journal of Mathematics and Its Applications
June 2024 Volume 18 Issue 2 Page 0785–0796
P-ISSN: 1978-7227 E-ISSN: 2615-3017
https://doi.org/10.30598/barekengvol18iss2pp0785-0796

DETERMINATION OF COFFEE FRUIT MATURITY LEVEL USING IMAGE HISTOGRAM AND K-NEAREST NEIGHBOR

Irene Devi Damayanti[1*], Aryo Michael[2]

[1,2]Study Program Informatics Engineering, Faculty of Engineering, Christian University of Indonesia, Toraja
Nusantara Street 12 Makale, South Sulawesi, 91811, Indonesia

Corresponding author's e-mail: ***irenedamayanti@ukitoraja.ac.id***

ABSTRACT

Article History:

Received: 28th October2023
Revised: 30th December 2023
Accepted: 4th March 2024
Published: 1st June 2024

Keywords:

Coffee Fruit Ripeness;
Digital Image Processing;
Image Histogram;
K-Nearest Neighbor;
RapidMiner.

Coffee has a very important role in the Indonesian economy, as one of the country's foreign exchange contributors in the plantation sector. Therefore, coffee processing is very important in determining the quality of coffee. The procedure for choosing and evaluating the coffee fruit's physical quality is one of the most crucial steps. The step of determining the maturity level of coffee fruit is carried out using the image histogram and K-Nearest Neighbor (KNN) method. This research uses the KNN algorithm with classification stages that will show the level of accuracy value according to the value of $k = 5$ used when processing the classification of coffee fruit image data. In order to complete this step, the features of the coffee fruit are identified using its color. The qualities of quality coffee fruit, which is flawlessly red in color. Twenty images total—ten of which are of ripe coffee fruit and ten of which are of raw coffee fruit—were used in this study. The test results were carried out using RapidMiner tools using 40% training data and 60% testing data from the total data set. Based on the test results, it gives an accuracy value of 100%, meaning that the data set can be used in the next stage as valid data to be used.

How to cite this article:

I. D. Damayanti and A. Michael., "DETERMINATION OF COFFEE FRUIT MATURITY LEVEL USING IMAGE HISTOGRAM AND K-NEAREST NEIGHBOR," *BAREKENG: J. Math. & App.*, vol. 18, iss. 2, pp. 0785-0796, June, 2024.

Research Article · Open Access

1. INTRODUCTION

As one of the country's main foreign exchange contributors from the plantation sector, coffee plays a very important role in the Indonesian economy. Therefore, coffee processing is very important to determine the quality of coffee because coffee production in Indonesia is currently still very low due to the low quality of the coffee fruit produced. One very important stage is the process of selecting and testing the physical quality of coffee fruit. Currently, this process is still done traditionally. In addition, there is insufficient data on the proper procedure for selecting coffee cherries, especially regarding the level of ripeness.

Almost all over the world, most people now drink coffee every day. Even today, coffee shops are increasingly popular. The quality of the coffee fruit also determines the flavor of the coffee. The quality of the coffee fruit is usually determined by its size, color, and texture. The characteristics of a good coffee fruit are that the color is perfect, the fruit is not empty, it is not attacked by insect pests, and the male and green coffee are separated from each other.

The coffee fruit image detection processing method can be used to determine the maturity level of coffee fruit. This is done by identifying the characteristics of the coffee fruit based on its color. Digital image processing is the method of processing digital images by computers through the use of appropriate algorithms [1], [2]. This is the basis of the research referred to as "Determination of Coffee Fruit Maturity Level Using Image Histogram and K-Nearest Neighbor Method." The purpose of this study is to create an image processing system that uses the color parameter of coffee skin to determine the maturity level of coffee fruit. This study was conducted with the aim of improving the accuracy of previous studies [3], [4], [5], and [6].

People can know the quality of coffee fruit and get information through a system that determines the maturity level of coffee fruit. This research uses the red, green, and blue (RGB) image method. Based on previous research, the average test results for determining the characteristic parameters of ripe coffee are visible, especially the mean RGB, are mostly dominated by red skin color. The same thing happens to the test results of the average calculation of the characteristic parameters of raw coffee, where the mean RGB is dominated by green skin color. The overall image processing process to indicate the maturity level of coffee fruit using a statistical approach method with image histograms achieved a success rate of 100% [7]. This research will be continued by applying a classification model using the K-Nearest Neighbor algorithm procedure. Where previously research has also been conducted on innovations to determine the maturity level of roasted coffee beans by utilizing RGB and HSV color features using the K-Nearest Neighbor algorithm and Principal Component Analysis algorithm. From this research, the accuracy result is 93.33% with the classification results of the test data sample as much as 28 data received accurate classification results and 2 data getting inaccurate classification results [8].

2. RESEARCH METHODS

2.1 Data Gathering

Data for the study was obtained from the search for ripe coffee fruit and raw coffee in Randanan Village, Tana Toraja Regency by selecting good and clean coffee fruit. Image capture of ripe and unripe coffee fruit was conducted at the Informatics Engineering Laboratory, Faculty of Engineering, Christian University of Indonesia, Toraja.

The first step is to take photos of raw and ripe coffee. Imaging was performed after the coffee fruit samples were placed on a plate. Next, a mobile phone camera was used to photograph the coffee fruit with a resolution of 2064 x 2064 pixels. The camera was placed 20 cm away from the ripe and unripe coffee fruit to take the photo. In order to classify coffee fruit types, the study used 20 sample images of coffee fruit—10 of which were of ripe coffee fruit and 10 of which were of unripe coffee fruit. The purpose of this study was to ensure that the coffee fruit images were sufficiently clear (no blur) and consistent for each shot.

After a photo of the coffee fruit is taken, the next step is image processing to obtain the RGB color component values. This is done in two steps, namely cropping and extraction of the average color component values of the ripe and unripe coffee fruit images. The cropping process is done to remove the dirty area of the object from the background.

After the cropping process, the next step is to calculate the average RGB color component value of each cropped image. This is done by extracting the red, green, and blue color component values for each pixel of the cropped image, and then calculating the average value of each red, green, and blue color component for each pixel. To determine the parameters of the percentage values of the RGB color components, the average values of the color components were extracted from the sample data of raw and cooked coffee images [10].

2.2 Simulation Design with Python

Python is a rapidly growing programming language that is widely recognized by programmers for its simple writing structure and language, which makes it easy to learn and understand. Because Python is an open-source project, anyone can build, add, expand, and use libraries for a variety of uses. Nowadays, Python is applied in numerous domains. Digital image processing often uses OpenCV [9].

In Figure 1 below shows how the simulation was designed using the Python programming language.

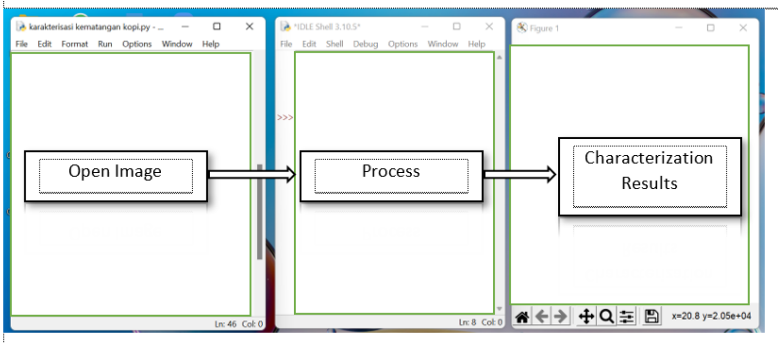

karakterisasi kematangan kopi.py - ...
File Edit Format Run Options Window Help
IDLE Shell 3.10.5
File Edit Shell Debug Options Window Help
>>>
Figure 1
Open Image
Process
Characterization Results
Ln: 46 Col: 0
Ln: 8 Col: 0
x=20.8 y=2.05e+04

Figure 1. Python Interface

2.3 RGB Image Histogram

A graph that can show the distribution of pixel intensity against the number of pixels at a certain intensity is called an image histogram. Therefore, histogram is a very important tool for qualitative and quantitative image management. Suppose a digital image has L gray degrees, which range from 0 to $L-1$. For example, the Equation (1) can be used to mathematically calculate the image histogram [11]:

$$h_i = \frac{n_i}{n};\ i\ =\ 0, 1, \dots, L-1 \tag{1}$$

where,
n_i = number of pixels that have gray degree i
n = the total number of pixels in the image

However, the histogram can be displayed graphically in the form of a bar chart, as shown in Figure 2. In Figure 2, the n_i values have been normalized by dividing by n, and the h_i values are between 0 and 1.

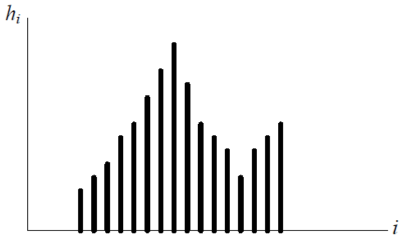


Figure 2. Image Histogram

In RGB color images, each pixel has its own color, which is indicated by the number of red, green, and blue colors present in it. The total number of possible different colors is 256^3, with three values for each pixel, if each component has a range from 0 to 255 [12]. It is very effective and accurate to compare two images with a color histogram [13].

One of the contrast enhancement operations based on manipulating the image is histogram equalization [14]. The main idea behind the histogram equalization problem is that we need to find a way to change the function $y = f(x)$. This transformation will map the gray scale images values from input to output and at the same time transform the PDF (Probability Density Function) distribution of the input $p_x(x)$ to produce the desired output PDF distribution $p_y(y)$.

Standardized results from the basis of probability theory state that:

$$p_y(y) = p_x(x)\left|\frac{dx}{dy}\right| \tag{2}$$

After that, take a look at the transformation function to calculate the integral, or area under the input probability density curve, between 0 and the upper limit of x:

$$y(x) = \int_0^x p_x(x')\,dx' \tag{3}$$

Retrieved by using Leibniz's rule and substituting it into the previous equation, we get:

$$p_y(y) = p_x(x)\left|\frac{1}{p_x(x)}\right| \tag{4}$$

Finally, since $p_x(x)$ is a probability density and is positive ($0 \le p_x(x) \le 1$), then:

$$p_y(y) = \frac{p_x(x)}{p_x(x)} = 1, \quad (0 \le y \le 1) \tag{5}$$

Thus, the output probability density $p_y(y)$ is constant, indicating that each output intensity y has the same probability. Therefore, the histograms of the output intensity values are the same.

Intensity values in digital images actually exist in only a limited number of levels and are discrete. We consider S possible discrete intensity values represented by x_k, where $k = \{0, 1, 2, \dots, S-1\}$, and the input probability density for level x_k is $p_x(x_k)$. The necessary general mapping $y = f(x)$ in this discrete case can be specifically defined for x_k using the following summation:

$$y(x_k) = \sum_{j=0}^{k} p_x(x_k) \tag{6}$$

The value of the kth entry of $y(x_k)$ is the sum of all entries in the histogram bin up to and including k. In essence, this is the cumulative histogram for the input image x.

Considering that the input and output image intensity values that are permitted can only be represented by a discrete integer value k, where $k = \{0, 1, 2, \dots, S-1\}$, it can be written as follows:

$$y(x_k) = \sum_{j=0}^{k} p_x(j) = \frac{1}{N}\sum_{j=0}^{k} p_x(n_j) \tag{7}$$

where the number of populations at the kth level is symbolized by n_k and the image's total number of pixels is N.

2.4 K-Nearest Neighbor (KNN)

A feature that allows for the differentiation of each class is necessary for class classification. The K-NN algorithm groups the outcomes of new query instances according to the majority of class labels present in KNN. It does this by using a supervised algorithm. The KNN algorithm groups testing data according to the k nearest neighbors of the training data. This is how it operates. One way to calculate the closest distance, namely using the Euclidean distance [15]. In Equation (8), the distance between the training data and the evaluated data is measured using the Euclidean distance [16], [17].

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \qquad (8)$$

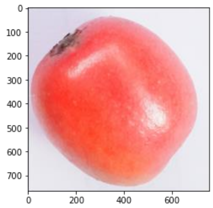
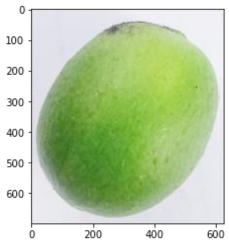
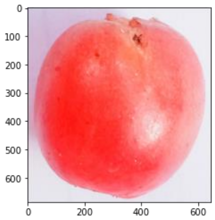
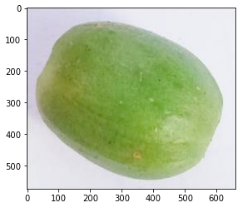
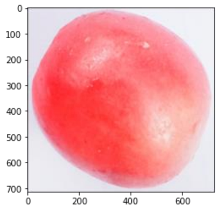
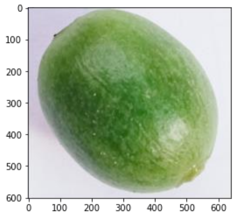
The K-NN algorithm starts by figuring out what is k. Next, it calculates the distance between every data point that needs to be assessed and all of the training data. The next step is to sort the distances that have been obtained before determining the closest distance. Finally, the number of classes of the nearest neighbor and the data class will be determined.

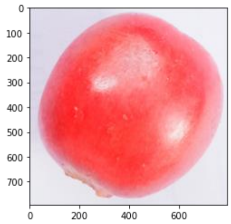
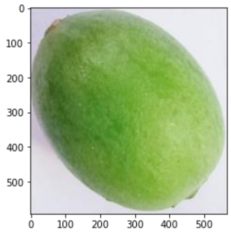
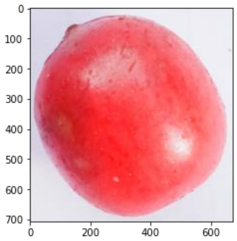
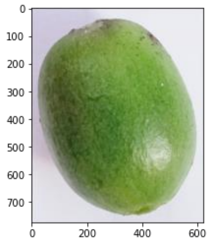
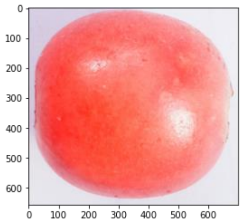
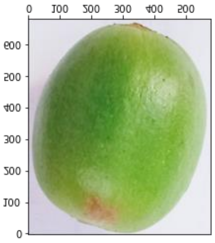
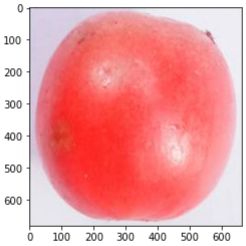
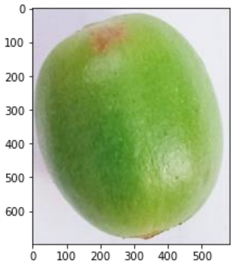
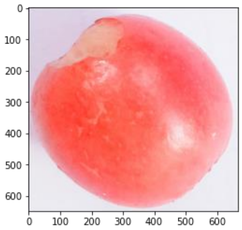
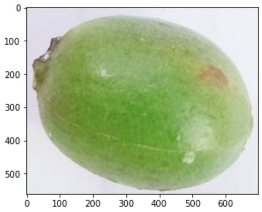
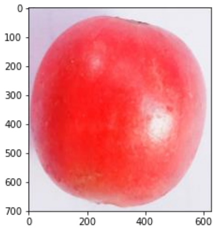
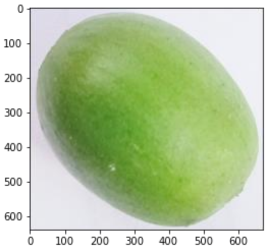
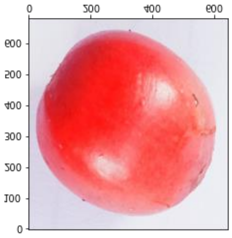
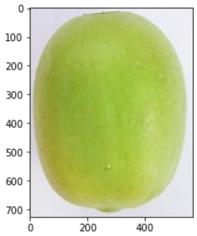
3. RESULTS AND DISCUSSION

3.1 Display Color Histogram (RGB)

This study used 20 samples of coffee fruit images, 10 of which were of ripe coffee fruit and 10 of which were of unripe coffee fruit, to determine the maturity level of coffee fruit. Images of coffee fruit maturity level are shown in Table 1.

Table 1. Image of Ripe Coffee Fruit and Raw Coffee

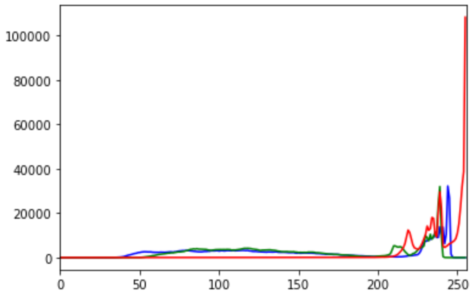
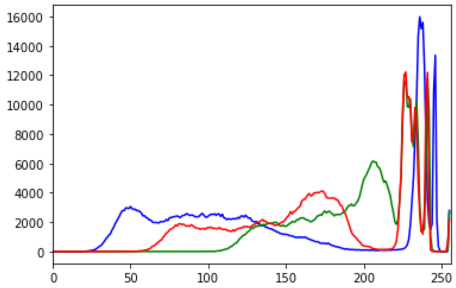
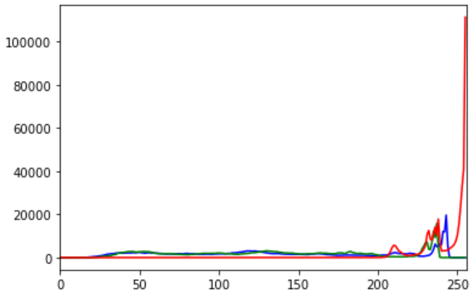
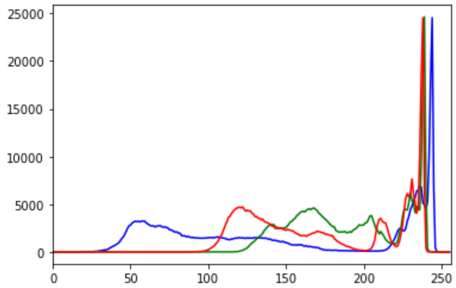
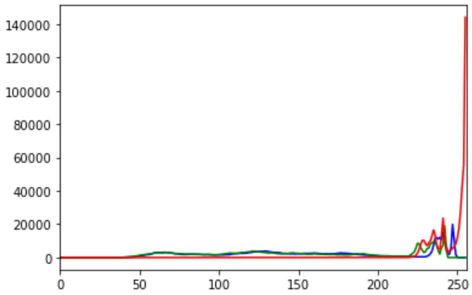
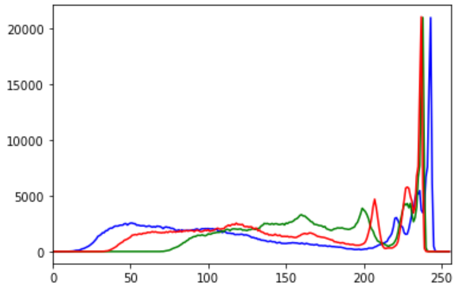
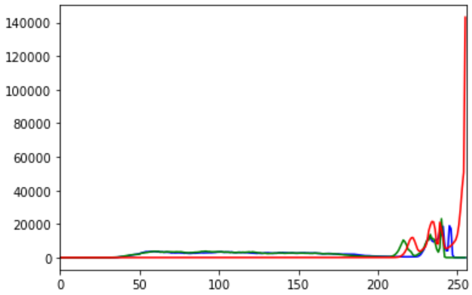
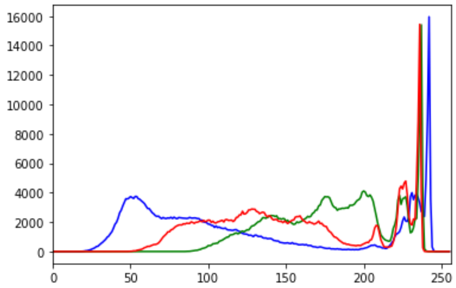
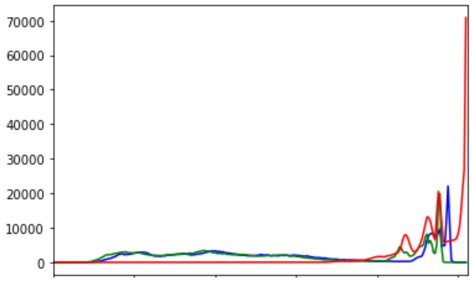
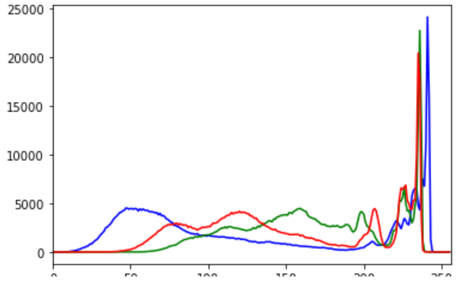
Number	Image of Ripe Coffee Fruits	Image of Raw Coffee Fruits
1.	 0 100 200 300 400 500 600 700 0 200 400 600	 0 100 200 300 400 500 600 0 200 400 600
2.	 0 100 200 300 400 500 600 0 200 400 600	 0 100 200 300 400 500 0 100 200 300 400 500 600
3.	 0 100 200 300 400 500 600 700 0 200 400 600	 0 100 200 300 400 500 600 0 100 200 300 400 500 600

4.		
5.		
6.		
7.		
8.		
9.		
10.		

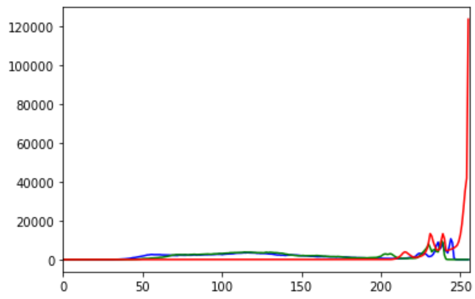
Before creating the main program, it is necessary to create a database to store image color calculation data that will be used as a comparison of test images. Thus, it can be known whether the test image includes images of unripe or ripe coffee cherries. For database creation, each image uses the color of ripe (red) and unripe (green) coffee fruit skin and calculates the mean parameter value. Python software stores this database in workspace storage.

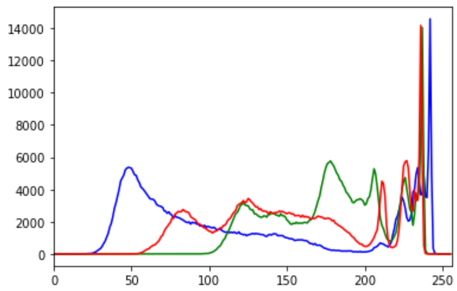
In **Table 2** below shows the results of the Histogram of Ripe Coffee Fruit and Raw Coffee Images,

Table 2. Histogram Results of Ripe Coffee Fruit and Raw Coffee Images

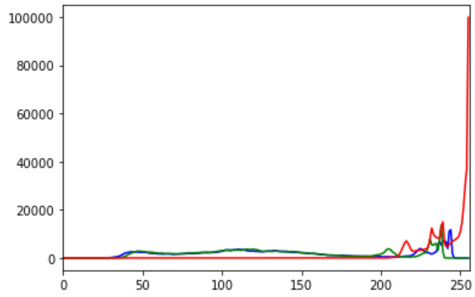
Number	Histogram of the Ripe Coffee Image	Histogram of the Raw Coffee Image
1.		
2.		
3.		
4.		
5.		

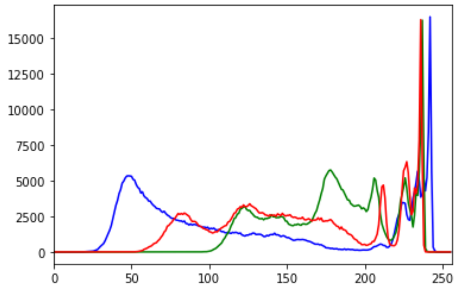
6.



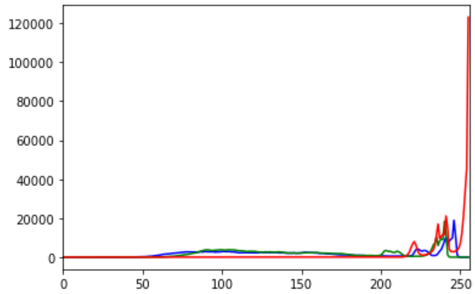


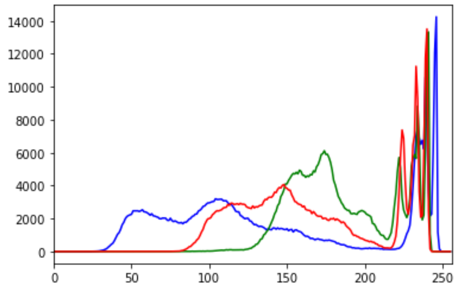
7.



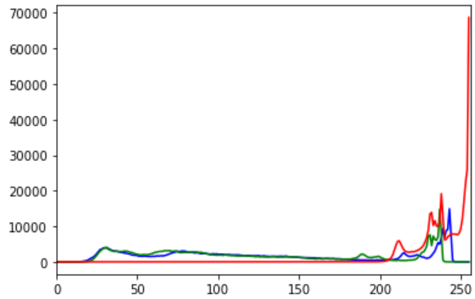


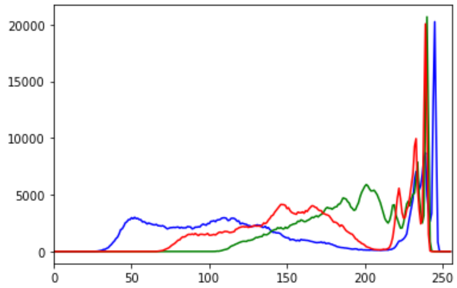
8.



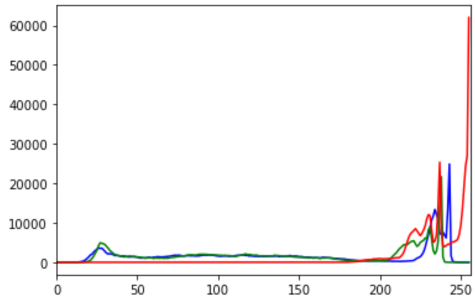


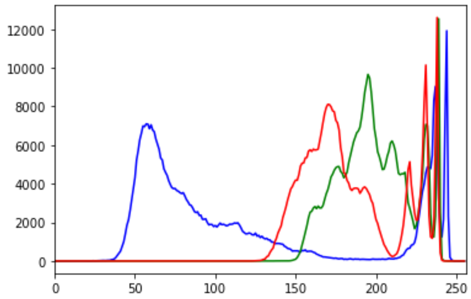
9.





10.





To determine the most prominent graphic color in each histogram, **Table 2** shows the scheme used to use red, green, and blue (RGB) histograms. To obtain RGB color pairs, RGB color pairs are used as initial cluster centers and cluster numbers are used to group each pixel into the corresponding area **[18]**.

3.2 Calculation of Mean Parameters

In **Table 3** below shows the calculation results of the mean parameter for the color of the ripe coffee fruit skin,

Table 3. Calculation results of RGB mean for ripe coffee images

Type of Coffee	RGB Mean Value of Ripe Coffee Image			Type of Coffee
	Red (R)	Green (G)	Blue (B)	
1	239.949	160.875	159.691	*Ripe Coffee*
2	243.366	139.835	142.641	*Ripe Coffee*
3	246.509	155.737	161.060	*Ripe Coffee*
4	243.111	150.443	156.965	*Ripe Coffee*
5	236.553	138.263	146.658	*Ripe Coffee*
6	245.183	147.662	145.010	*Ripe Coffee*
7	242.767	138.325	141.656	*Ripe Coffee*
8	245.655	159.261	159.472	*Ripe Coffee*
9	239.719	124.526	131.012	*Ripe Coffee*
10	237.713	145.779	150.746	*Ripe Coffee*

In **Table 4** below shows the calculation results of the mean parameter for the color of raw coffee fruit skin,

Table 4. Calculation results of RGB mean for raw coffee images

Type of Coffee	RGB Mean Value of Raw Coffee Image			Type of Coffee
	Red (R)	Green (G)	Blue (B)	
1	173.361	198.778	145.773	*Raw Coffee*
2	151.910	192.002	176.583	*Raw Coffee*
3	153.898	174.278	139.586	*Raw Coffee*
4	154.649	181.812	125.414	*Raw Coffee*
5	153.165	170.527	131.017	*Raw Coffee*
6	156.282	179.644	123.710	*Raw Coffee*
7	157.881	180.843	126.125	*Raw Coffee*
8	170.589	188.860	144.462	*Raw Coffee*
9	172.404	194.609	144.862	*Raw Coffee*
10	184.813	199.037	120.365	*Raw Coffee*

It is evident that the average test results of the computation of the mean RGB parameters of images of ripe coffee are dominated by the color red, as shown in **Table 3**. Likewise, in **Table 4**, green skin tone predominates in the average test results of the computation of the mean RGB parameters of raw coffee images.

3.3 K-NN Algorithm Classification with RapidMiner

The classification model is carried out through the K-NN algorithm process which consists of the read excel operator to accommodate the data set, the split data operator to set the training data by 40% and the testing data by 60% of the total data set contained in the read excel operator, the K-NN operator is the process form of the K-NN algorithm itself, the apply model operator is used to combine the K-NN and split data operators into a single model, and the performance operator to measure the percentage level of performance accuracy of the K-NN model that has been formed.

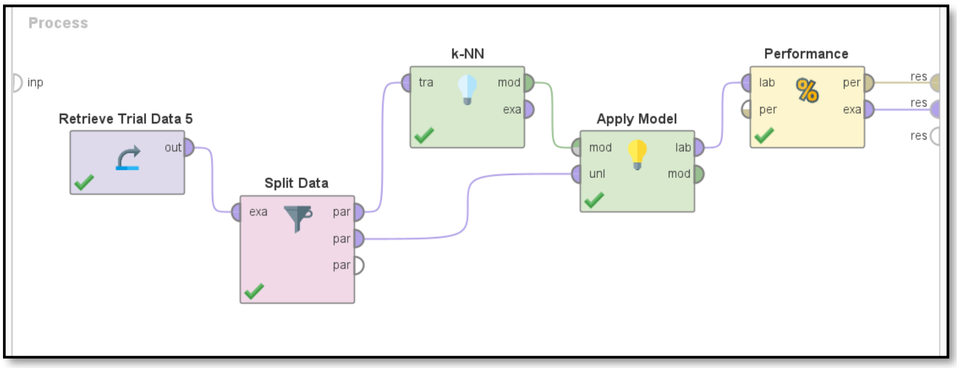
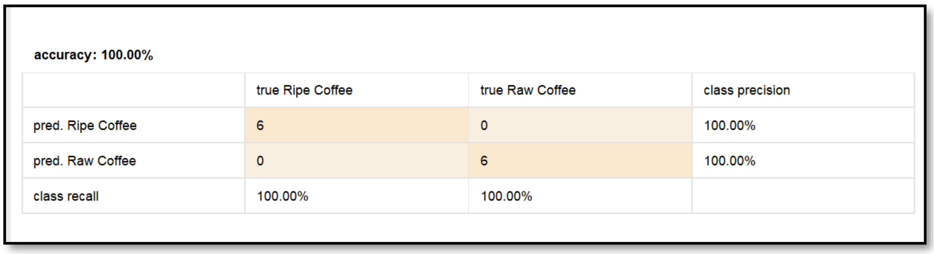


Figure 3. **K-NN Classification Model**

3.4 Testing with RapidMiner

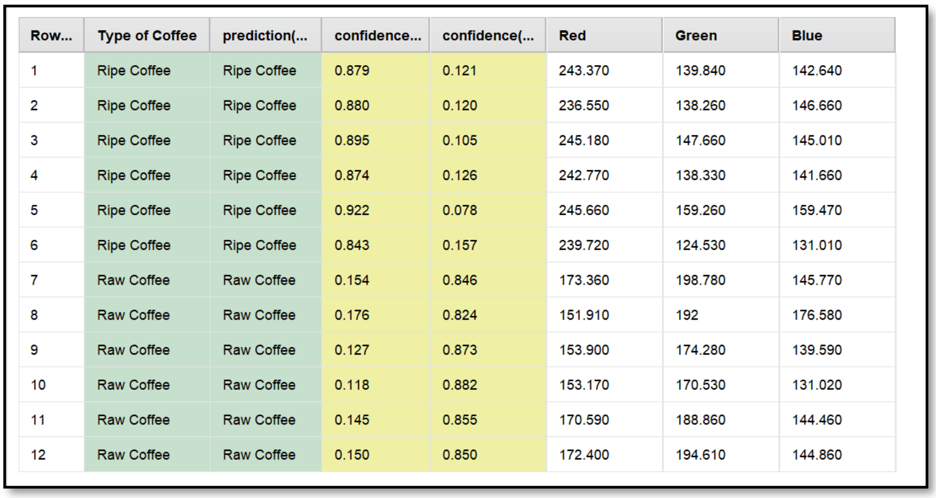
This research uses the K-NN algorithm with classification stages that will show the level of accuracy value according to the $k = 5$ value used when processing the classification of coffee fruit image data. The value of $k = 5$ means that there are 5 adjacent vectors that will be used as a comparison to be able to represent feature vectors from various classes [19]. The test results with RapidMiner are shown in Figure 4 and Figure 5 below,



accuracy: 100.00%

	true Ripe Coffee	true Raw Coffee	class precision
pred. Ripe Coffee	6	0	100.00%
pred. Raw Coffee	0	6	100.00%
class recall	100.00%	100.00%	

Figure 4. **Testing Accuracy Value with RapidMiner**



Row...	Type of Coffee	prediction(...	confidence...	confidence(...	Red	Green	Blue
1	Ripe Coffee	Ripe Coffee	0.879	0.121	243.370	139.840	142.640
2	Ripe Coffee	Ripe Coffee	0.880	0.120	236.550	138.260	146.660
3	Ripe Coffee	Ripe Coffee	0.895	0.105	245.180	147.660	145.010
4	Ripe Coffee	Ripe Coffee	0.874	0.126	242.770	138.330	141.660
5	Ripe Coffee	Ripe Coffee	0.922	0.078	245.660	159.260	159.470
6	Ripe Coffee	Ripe Coffee	0.843	0.157	239.720	124.530	131.010
7	Raw Coffee	Raw Coffee	0.154	0.846	173.360	198.780	145.770
8	Raw Coffee	Raw Coffee	0.176	0.824	151.910	192	176.580
9	Raw Coffee	Raw Coffee	0.127	0.873	153.900	174.280	139.590
10	Raw Coffee	Raw Coffee	0.118	0.882	153.170	170.530	131.020
11	Raw Coffee	Raw Coffee	0.145	0.855	170.590	188.860	144.460
12	Raw Coffee	Raw Coffee	0.150	0.850	172.400	194.610	144.860

Figure 5. **Testing Results with RapidMiner**

In Figure 4, shows the results of testing with RapidMiner resulting in an accuracy value of 100%, meaning that the data set can be used in the next stage as valid data for use.

The test results using the RapidMiner analysis tool, by dividing the data into 40% for training and 60% for testing from the entire dataset, showed that the classification model managed to achieve an accuracy value of 100%. These results indicate the success of the model in classifying the data at the testing stage, and this dataset can be considered as valid and reliable data for the next stage of the research. This research also revealed the success of the classification model implemented using the K-Nearest Neighbor algorithm. This increase in accuracy reflects progress in the accuracy of the research results compared to previous studies.

Therefore, this method can be considered as a more effective approach in the context of classifying data in this study. This increase in accuracy can have a positive impact on the validity and reliability of the research results, strengthening confidence in the conclusions derived from the data analysis.

4. CONCLUSIONS

Based on the test results conducted using RapidMiner tools using 40% training data and 60% testing data from the total data set, it gives an accuracy value of 100%, meaning that the data set can be used in the next stage as valid data to use.

REFERENCES

[1] D. S. Shreya, “Digital Image Processing and Recognition Using Python,” *Int. J. Eng. Appl. Sci. Technol.*, vol. 5, no. 10, pp. 319–322, 2021, doi: 10.33564/ijeast.2021.v05i10.046.

[2] A. McAndrew, A Computational Introduction to Digital Image Processing, Second Edition, Melbourne, Australia: CRC Press, 2016.

[3] Z. K. Simbolon, S. A. Syakry, Mulyadi, and M. Syahroni, “Separation of the Mature Level of Papaya Callina Fruit Automatically Based on Color (RGB) uses Digital Image Processing,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 536, no. 1, 2019, doi: 10.1088/1757-899X/536/1/012127.

[4] T. Meruliya, P. Dhameliya, J. Patel, D. Panchal, P. Kadam, and S. Naik, “Image Processing for Fruit Shape and Texture Feature Extraction - Review,” *Int. J. Comput. Appl.*, vol. 129, no. 8, pp. 30–33, 2015, doi: 10.5120/ijca2015907000.

[5] Y. Guan, F. Zhou, and J. Zhou, “Research and Practice of Image Processing Based on Python,” *J. Phys. Conf. Ser.*, vol. 1345, no. 2, 2019, doi: 10.1088/1742-6596/1345/2/022018.

[6] D. Iskandar and Marjuki, “Classification of Melinjo Fruit Levels Using Skin Color Detection With Rgb and Hsv,” *J. Appl. Eng. Technol. Sci.*, vol. 4, no. 1, pp. 123–130, 2022, doi: 10.37385/jaets.v4i1.958.

[7] I. D. Damayanti, “Karakterisasi Tingkat Kematangan Buah Kopi Menggunakan Pendekatan Statistik,” *J. Sains dan Sist. Teknol. Inf.*, vol. 5, no. 2, pp. 119–127, 2023, doi: 10.59811/sandi.v5i2.62.

[8] M. I. Pratama, . K., and A. H. Muhammad, “Classification of coffee beans roast maturity levels based on digital image processing color using the KNN and PCA method,” *Int. J. Sci. Res. Publ.*, vol. 12, no. 12, pp. 138–148, 2022, doi: 10.29322/ijsrp.12.12.2022.p13217.

[9] A. B. Abadi and S. Tahcfulloh, “Digital Image Processing for Height Measurement Application Based on Python OpenCV and Regression Analysis,” *Int. J. Informatics Vis.*, vol. 6, no. 4, pp. 763–770, 2022, doi: 10.30630/joiv.6.4.1013.

[10] I. D. Damayanti, A. Michael, H. K. Y. Piopadang, and P. Setriyanti, “Klasifikasi Citra Daging Babi dan Daging Kerbau Menggunakan Histogram Citra dan GLCM,” vol. 4, no. 2, pp. 188–200, 2023.

[11] R. Munir, Pengolahan Citra Digital dengan Pendekatan Algoritmik, Bandung: Penerbit Informatika, 2004.

[12] H. P. Hanusch, "Digital Image Processing Using Matlab", Institute of Geodesy and Photogrammetry, ETH Zurich.

[13] M. Pydi and K. L. Sailaja, “A Framework for the Image Retrieval System Based on Histogram Normalization Technique with Python,” *Int. J. Eng. Adv. Technol.*, vol. 9, no. 4, pp. 2300–2304, 2020, doi: 10.35940/ijeat.d9060.049420.

[14] A. Vyas, S. Yu, and J. Paik, *Fundamentals of digital image processing*. 2018. doi: 10.1007/978-981-10-7272-7_1.

[15] A. W. Satria Bahari Johan, S. W. Putri, G. Hajar, and A. Y. Wicaksono, “Modified KNN-LVQ for Stairs Down Detection Based on Digital Image,” *Lontar Komput. J. Ilm. Teknol. Inf.*, vol. 12, no. 3, p. 141, 2021, doi: 10.24843/lkjiti.2021.v12.i03.p02.

[16] S. Anraeni, D. Indra, D. Adirahmadi, S. Pomalingo, Sugiarti, and S. H. Mansyur, “Strawberry Ripeness Identification Using Feature Extraction of RGB and K-Nearest Neighbor,” *3rd 2021 East Indones. Conf. Comput. Inf. Technol. EIConCIT 2021*, pp. 395–398, 2021, doi: 10.1109/EIConCIT50028.2021.9431854.

[17] S. Anraeni and Herman, “Hybrid lacunarity and euclidean distance algorithms for kidney health classification through iris image,” *Int. J. Sci. Technol. Res.***,** vol. 8, no. 11, pp. 486-488, 2019.

[18] S. Basar, M. Ali, G. Ochoa-Ruiz, M. Zareei, A. Waheed, and A. Adnan, “Unsupervised color image segmentation: A case of RGB histogram based K-means clustering initialization,” *PLoS One*, vol. 15, no. 10 October, pp. 1–21, 2020, doi: 10.1371/journal.pone.0240015.

[19] M. Effendi, M. Jannah, and U. Effendi, “Corn quality identification using image processing with k-nearest neighbor classifier based on color and texture features,” *IOP Conf. Ser. Earth Environ. Sci.*, vol. 230, no. 1, 2019, doi: 10.1088/1755-1315/230/1/012066.