Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (143-156)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.41225

JURNAL TEKNIK INFORMATIKA

Homepage : http://journal.uinjkt.ac.id/index.php/ti

Classification of Coconut Fruit Quality Using The K-Nearest Neighbour (K-NN) Method Based on Feature Extraction: Color, Shape, and Texture

Sucinda Kardena[1*], Fildza Izzati[2], and Rusdah[3]

[1,2,3]Master of Computer Science, Faculty of Information Technology, Budi Luhur University
[1,2,3]Jl. Ciledug Raya, South Jakarta, Indonesia

Article:
Accepted: January 03, 2025
Revised: August 26, 2024
Issued: April 30, 2025

***Correspondence Address:**
2111602161@student.budiluhur.ac.id

ABSTRACT

In 2021, Indonesia was the world's largest coconut producer, with production reaching 17.1 million tons, according to the Food and Agriculture Organization (FAO). However, due to the long distribution time from farmers to consumers, the quality of coconuts often decreases, mainly due to manual classification. Coconuts that meet consumption standards are considered suitable, while coconuts that are overripe, damaged, or unripe are considered Non-standard. To overcome this problem, an automatic classification system was developed using machine learning with the K-Nearest Neighbor (K-NN) algorithm. The total required dataset is around 500, comprising 250 standard coconut datasets and 250 non-standard coconut datasets. The dataset was taken from coconut Images from Indragiri Hilir, Riau Province. Coconut features colour, shape, and texture.. The development process used the Cross Industry Standard Process for Data Mining (CRISP-DM). The evaluation used a confusion matrix .This study explores five training-test ratio data split scenarios of 90:10, 80:20, 70:30, 60:40, and 50:50. The highest accuracy, 96%, is achieved with a data split of 90:10 and a K value 5. Then, the K-NN model will be compared with other models, for Support Vector Machine (SVM) with RBF kernel accuracy of 94%, SVM with Linear kernel of 90%, Random Forest with accuracy of 92%, and Convolutional Neural Network (CNN) with accuracy of 86%.

Keywords : *coconut; k-nearest neighbour (K-NN); Image processing; feature extraction; confusion matrix.*

Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (143-156)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.41225

1. INTRODUCTION

Coconut (Cocos nucifera L.) is a vital agricultural commodity with significant social, cultural, and economic importance in Indonesia [1]. Due to its extensive applications, ranging from food and cosmetics to industrial products, it is regarded as a multipurpose crop, particularly in coastal communities. According to the Food and Agriculture Organization (FAO), Indonesia was the world's largest coconut producer in 2021, with a total production of 2.85 million tons [2]. The majority of this production originates from Riau Province, benefiting from Indonesia's favorable climate and geographical conditions, which enable an annual production exceeding 18 million tons.

Coconut quality is a key factor influencing its market value. Proper classification ensures higher pricing and enhances consumer confidence, as it distinguishes between standard and non-standard coconuts. For instance, as of August 6, 2023, the price of standard coconuts ranged from 1,300 to 1,400 rupiah per kilogram, whereas non-standard coconuts were valued between 800 and 900 rupiah per kilogram.

Currently, coconut classification is predominantly performed manually based on tactile inspection. Standard coconuts, which are free from defects such as sprouts, rot, cracks, or splits, are designated for direct consumption, whereas non-standard coconuts are typically processed into copra. However, this manual approach is labor-intensive, time-consuming, and prone to inconsistencies.

The dataset used is a coconut dataset; the dataset was taken using standards starting from the distance and the brightness and background of the coconut. This is done because the distance of the shot, the light and the background greatly influence the dataset.[3]

Several studies have explored the integration of information technology in coconut classification. Prior research has investigated quality assessment after the peeling process using Image processing and machine learning [4], efficient classification methods for coconuts [5], and maturity-based coconut detection [6].

Widians [7] , applied the K-Nearest Neighbors (K-NN) method for classification based on shape and texture feature extraction. Shape features were analyzed using metric and eccentricity parameters, while texture features were extracted using the Gray Level Co-occurrence Matrix (GLCM), considering contrast, correlation, energy, and homogeneity. Although this study focused on onion classification, it demonstrated the effectiveness of feature extraction and machine learning for agricultural product sorting, which is relevant to coconut classification

In research conducted by Hadi and Rachmawanto [8] , by applying digital Image processing technology using the RGB color feature extraction method, GLCM texture feature extraction and applying the K-Nearest Neighbor algorithm for the classification process. This research uses a dataset with a total of 260 Images used which will be divided into 4 classes, namely raw, half-cooked, mature and rotten Images. The highest accuracy is produced by K=1 at 98.75%.

Then in research conducted by Wijaya, Koredianto and Saidah [9] which discussed the classification of cattle types using the Gray Level Cooccurrence Matrix (GLCM) method using the K-Nearest Neighbor and Support Vector Machine (SVM) classification. In this test, an accuracy of 100% was obtained in the K-NN classification with a computing time of 0.967 s and in the SVM classification an accuracy level of 80% was obtained with a computing time of 1,570 s.

A study [10] comparing KNN and SVM in mango ripeness classification based on HSV Images and statistical features showed that KNN provided superior and more stable performance than SVM. In an experiment with 80 mango Images, KNN achieved 98.75% accuracy and 100% precision, while SVM with a linear kernel only achieved 97.50% accuracy and 97.78% precision. A study comparing KNN and SVM in mango ripeness classification based on HSV Images and statistical features showed that KNN provided superior and more stable performance than SVM. In an experiment with 80 mango Images, KNN achieved 98.75% accuracy and 100% precision, while SVM with a linear kernel only achieved 97.50% accuracy and 97.78% precision.

There has been a lot of research related to the coconut industry and information technology, including Image processing and AI 1). Calculation coconut trees at a location [11] [12] , 2) Selection of good sugar [13][14], 3). Detection of coconut diseases [15], 4) Separation of standard and non-standard

coconuts [16] and 5). Copra separation [17]. Apart from that, other research is related to separation Standard and non-standard coconuts have also been used for sound [18]and smell [19]

Building upon previous research, this study aims to develop an automated coconut classification system based on color, shape, and texture feature extraction. The K-Nearest Neighbors (K-NN) algorithm will be employed for classification, utilizing RGB and grayscale feature extraction, shape analysis through metric and eccentricity parameters, and texture assessment using the Gray Level Co-occurrence Matrix (GLCM) [20] [21], [22] [23]. This research is expected to enhance the efficiency and accuracy of coconut quality assessment, contributing to the development of precision agriculture in Indonesia.

2. METHODS

This study employs an experimental research method with a quantitative approach to evaluate the performance of the K-Nearest Neighbor (K-NN) model in classifying standard and non-standard coconuts based on extracted visual features.

2.1. Sample Sorting

In this research The primary dataset in this study consists of coconut Images collected through direct photography. To ensure dataset standardization, all Images were captured under controlled conditions. The coconuts were placed on a white HVS paper background before photography to maintain uniformity. The Images were captured using mobile phone cameras with a fixed distance of 50 cm between the camera and the coconut.

The dataset contains 500 Images of 250 standard and 250 non-standard coconut Images. The Image is a colour Image with the same length and width because standardization was carried out during the shooting process. However, the Image file size may vary because the Image was taken using more than one mobile phone device.

The 500 Images will later be split into training and testing data in the preprocessing process. Training data is the data used to train the model, and testing data is the data that will be used to measure the model's accuracy.

The number of datasets of 500 Images is considered sufficient based on research [24], which states that small samples can be adequate for research when the data set has high-quality data.

2.2. Design Techniques

The design technique in this research is to use an Image processing methodology which consists of: Image Acquisition, Image Processing, Feature Extraction, Split Dependent and Independent Value, Feature Scaling, Splitting Data, and Classification.

a. Image Acquisition

Image acquisition is photographing an object, namely a coconut, to obtain a coconut Image in digital form. The device that will be used to photograph coconuts is a camera with a distance between the cell phone and the object of 50 cm. The background for the object is HVS paper to make Image processing easier. The coconut Images consist of 250 standard coconut Images and 250 non-standard coconut Images. Standard coconuts are coconuts of good quality, while non-standard coconuts are coconuts that are broken, tiny, too young, too old and growing buds.

b. Image Processing

Image processing refers to the computational techniques applied to enhance Image quality and extract meaningful features for analysis. The Image processing stage includes labelling, resizing, background removal, and cropping to ensure uniform Image representation.

c. Feature Extraction

At this stage, digital Image data is converted into numerical data or numerical representation, which will later be used by data processing or machine learning algorithms. The feature extraction process in this study includes three primary components: color, shape, and texture extraction. For colour feature extraction, colour moments will be used, including the RGB and Greyscale mean. Shape feature extraction involves the use of metric and eccentricity parameters to characterize coconut morphology. Moreover, we will use the Gray Level Co-occurrence Matrix (GLCM) with 1 and 0 to extract texture features. For GLCM, the features that will be analysed are contrast, dissimilarity, homogeneity, energy and correlation. The results of this feature extraction will be used as parameters for the classification process using the K-Nearest Neighbour (K-NN) algorithm.

The research [20] shows that combining colour feature extraction, such as RGB and HSV, texture feature extraction, such as GLCM, and shape feature extraction, such as metric and eccentricity parameters, provides high accuracy.

d. Split Dependent and Independent Value

At this stage, data is separated between dependent data and independent data. The dependent value is also known as the target variable, output or response, which is the value that will be predicted as the output of machine learning. The dependent value in this research is the quality class of coconut fruit, namely standard and non-standard. Meanwhile, independent values, also known as features, predictors or input, are variables used as input for the model, where the model will use the attributes or information from the independent value to carry out predictions or analysis. Moreover, in this research, the independent value is the variable resulting from feature extraction. The attributes meanR, meanG, meanB, and mean grayscale will be produced for colour feature extraction. For feature extraction, size will produce metric and eccentricity attributes. Meanwhile, extracting Gray Level Co-occurrence Matrix (GLCM) features will produce contrast, dissimilarity, homogeneity, energy and correlation attributes.

e. Feature Scaling

Feature Scaling is normalizing or standardizing feature values (independent values) in a dataset before using them in the K-Nearest Neighbors (K-NN) algorithm. Feature scaling ensures that all extracted features are within a uniform range, preventing any single feature from disproportionately influencing the model due to differing value scales. This study employs the Min-Max Normalization method, which scales feature values within a predefined range [0,1] to maintain consistency across all extracted attributes.

f. Splitting Data

The next stage is splitting the data, where the dataset will be divided into two categories: testing data and training data. The dataset is partitioned into training and testing subsets using various ratios (60:40, 70:30, 80:20, and 90:10) to evaluate model performance across different data distributions. This is done to find what proportion and k value has the highest value and does not result in underfitting or overfitting of the model.

g. Classification

Classification is a supervised learning approach that categorizes data based on predefined attributes. This study utilizes the K-Nearest Neighbor (K-NN) algorithm for classification with K= 1, 3, 5, 7, and 9. which assigns class labels based on the majority vote of the k-nearest data points. K-NN classifies objects based on the similarity of characteristics with the closest or similar learning data. The distance between objects is calculated with the Euclidean metric, and this technique is simple but provides exemplary accuracy.

After performing K-NN classification, we will compare the best K-NN model with other models, namely Support Vector Machine (SVM), Random Forest, and Convolutional Neural Network (CNN).

2.3. Testing Techniques

Model performance is evaluated using the Confusion Matrix, measuring Accuracy, Precision, Recall, and F1-Score to assess classification effectiveness

2.4. Deployment

The model with the best accuracy will proceed to the deployment stage. Namely, it will be embedded in a web application that can automatically classify coconut quality.

3. RESULTS AND DISCUSSION

3.1. Image Acquisition

Coconut datasets are collected directly from the coconut production centre in Riau Province, specifically in the Indragiri downstream Regency. Photography standards include the exact distance between the object and the camera (for example, 50 cm), the use of a white background, and the use of a holder to maintain distance, stable lighting and stability when taking photos. The total number of required datasets is around 500, comprising 250 standard coconut datasets and 250 non-standard coconut datasets. The dataset was stored in a cloud repository for ease of access and processing.

Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (143-156)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.41225

Figure 1. *Example of a standard coconut dataset*

Figure 2. *Example of a nonstandard coconut dataset*

This step aims to gain insight from the data and evaluate the quality of the dataset by checking whether the entire dataset contains coconut Images or whether there are inappropriate Images. The technique used in checking coconut Images is to look at the data set Images individually. The data entered into the dataset is Image data that contains coconut Images and has a coconut-quality label

3.2. Preprocessing Data

This stage is applied to process Images before being used in modelling. In this research, Image processing involves four steps, namely labelling, resizing, removing background, and cropping.

a. Labelling

In the first stage of the labelling process, the author grouped the data by creating Standard Coconut and Non-Standard Coconut folders and entered the Standard Coconut Image from the data acquisition results on Google Drive into the folder. Coconut farmers assisted in this stage to ensure the coconuts in the folder corresponded to their class. Farmers carry out labeling based on guidelines from ICAO (International Civil Aviation Organization) [25] which explains the criteria for non-standard coconuts used in Bioavtur.

b. Resize

The resizing process is carried out by changing the Image size to 800×600 pixel, The selection of 800 x 600 pixel is based on creating a standard Image size so that the Image can be processed efficiently [26]. The resize results are then saved in the folder that was created previously.

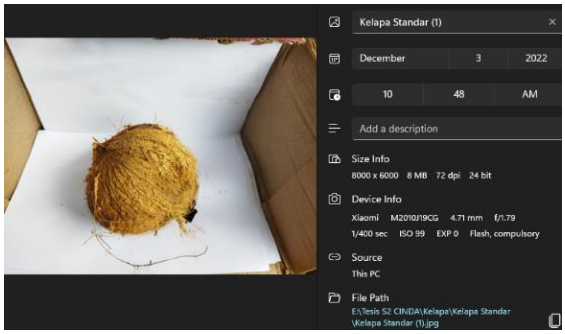

Figure 3. *Image before resizing*

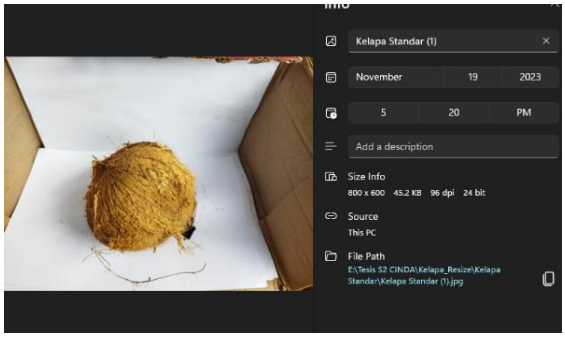

Figure 4. *Image after resizing*

The purpose of resizing the Image in Image processing is for computational efficiency, memory management, speeding up the model. in classification and making data more consistent.

c. Remove Background

The aim of this stage is to remove the background from the Image. After background removal, the Images were converted from JPG to PNG format to preserve transparency. remove background is done using Open CV.

Figure 5. *Image before removing background*

Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (143-156)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.41225

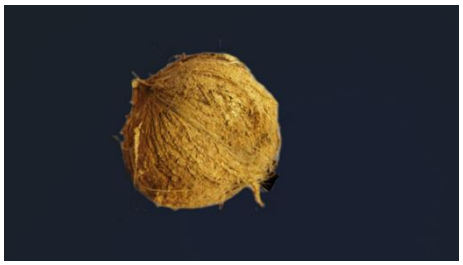

Figure 6. *Image after removing background*

Removing background is done in order to increase classification accuracy, reduce the impact of disturbing background and class separation will be clearer.

d. Cropping

The next step in data preprocessing is the Cropping stage. The first stage in the Image cropping process is to create the directory needed to save the results of the resizing process. Cropping allows the Model to focus on the essential features of the object, ignoring information that may be irrelevant. This helps in making classification decisions based on the most discriminative features. Cropping is done automatically using the Bounding Box Cropping technique.

3.3. Feature Extraction

a. Colour Feature Extraction

The first step is to extract RGB features, which are red, green, and blue. These three colours are the primary colours that form other colours written in values in the form of RGB triplets. Each component is usually given a value from 0 to a maximum 255. For example, the colour White has a composition of red = 255, green = 255 and Blue = 255, while the colour Black has the values red = 0, green = 0 and Blue = 0. The following is an example of the calculation against 3 x 3 RGB Images, which can be used to find the characteristic value of a colour using manual calculations.

R : 70 G : 71 B : 66	R : 75 G : 76 B : 70	R : 82 G : 82 B : 75
R : 87 G : 87 B : 79	R : 93 G : 92 B : 82	R : 96 G : 94 B : 84
R : 97 G : 79 B : 84	R : 100 G : 98 B : 87	R : 101 G : 99 B : 86

Figure 7. *Example of RGB 3×3*

The following is the process of finding the mean value based on the 3x3 RGB Image value components using the equation:

Mean R =1/9∑R
Mean G =1/9∑G
Mean B =1/9∑B

The following is an example of the calculation

MeanR =1/9 (70+75+82+87+93+96+97+100+101)
= 89

MeanG =1/9 (71+76+82+87+92+94+79+98+99)
= 86.44

MeanB =1/9 (66+70+75+79+82+84+84+87+86)
=78.11

Based on the results of calculations to find the mean value of RGB colour moments, three colour characteristic values were obtained, namely: MeanR = 89, MeanG = 86.44 and MeanB = 78.11

The next step is to extract HSV colour characteristics in an Image. This can be done by converting it from BGR (default from OpenCV) to an HSV Image. Next, take the mean value for each Image. This means HSV value will be used as a colour characteristic of the copra extracted type. For example, a 3×3 HSV Image can be used to find the characteristic values of coconuts.

H : 36 S : 16 V : 81	H : 35 S : 18 V : 86	H : 30 S : 19 V : 92
H : 30 S : 21 V : 97	H : 27 S : 27 V : 103	H : 25 S : 29 V : 106
H : 30 S : 31 V : 107	H : 25 S : 30 V : 110	H : 26 S : 34 V : 111

Figure 8. *HSV 3×3 image*

The following is a calculation process to find the mean value based on the HSV color Image value components.

MeanH =1/9 (36+35+30+30+27+25+30+25+26)
= 29.33

MeanS =1/9 (16+18+19+21+27+29+31+30+34)
= 25

MeanV =1/9 (81+86+92+97+103+106+107+110+111)
= 99.22

Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (143-156)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.41225

Based on the calculation results to find the mean value of the HSV colour moment, three colour characteristic values were obtained: MeanH = 29.3333, MeanS = 25, and MeanV = 99.222.

Next, Grayscale colour characteristics are extracted from an Image by converting it from BGR (default from OpenCV) to a Grayscale Image. The following is an example of a calculation to obtain a grayscale Image.

$GrayScale11 = (0.3 \times 70) + (0.59 \times 71) + (0.11 \times 66)$
$GrayScale11 = 70.15$

This Grayscale mean will be used as a colour characteristic of the copra-extracted type. For example, there is a 3×3 Greyscale Image.

70	75	81
86	91	93
96	97	98

Figure 9. *Grayscale 3x3 image*

Figure 9 shows that all RGB values are calculated using a formula

Grayscale = (0.3R) + (0.59G) + (0.11B).

Then carry out the Mean calculation process based on the Grayscale Image value with the equation:

Mean Grey Scale =1/9∑Grayscale = 87.44

Then, calculate the standard deviation and variation of the Grayscale values.

Standard Deviation = 48.94838
Variation = 2395.29167

b. Shape Feature Extraction

The form extraction used in this research is Area and Perimeter. In this case the author uses the example of a grayscale Image.

80	85	91	25	50
96	101	103	75	111
106	150	200	96	91
70	135	167	104	35
50	100	120	83	89

Figure 10. *Grayscale 5×5 image*

Next, carry out thresholding with a threshold value of 127. The results of this thresholding operation will produce a binary Image where pixels that initially have a value above 127 will become white (value 255) to represent the object, while pixels that have a value below or equal to 127 will be black (value 0) to represent the background.

0	0	0	0	0
0	0	0	0	0
0	255	255	0	0
0	255	255	0	0
0	0	0	0	0

Figure 11. *5×5 binary image*

Each line segment that forms a contour will contribute to the area and perimeter of the contour. The most prominent contour's area and perimeter length (perimeter) will be calculated if a contour is found. If no contour is found, the area and perimeter will be 0.

In this case, the contour is found with a value of 255 in the middle box. Then, look for the most prominent contour. In the example given, only one contour was found, and this contour is used as the most prominent contour. This contour is selected as the central contour in the object. Next, the area and perimeter values are calculated.

Area = number of pixels with a value of 255 on the most giant contour = 4 pixels

Perimeter = 2 × (number of horizontal pixels + number of vertical pixels) = 2 × (2 + 4) = 12 pixels

c. GLCM Feature Extraction

The feature extraction used in this research is the GLCM (Gray Level Co-occurrence Matrix) method with several features such as contrast, dissimilarity, homogeneity, energy and correlation. The following are the steps for manual calculations:

1. In this case, using the example of a 3×3 matrix.
2. Determines the spatial relationship between pixels by initiating distance and angle values. In this case, a distance of 1 and an angle of 0° are used.
3. Create a GLCM matrix for each pixel by calculating how often pairs of pixels appear at a specified distance and direction.

Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (143-156)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.41225

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0
4	0	0	0	2	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	2	0	0	0
7	0	0	0	0	0	0	1	0

Figure 12. *GLCM matrix*

4. Normalize the matrix to get its probability distribution.

This is done by dividing each matrix element by the total number of pixel pairs counted

Calculating the GLCM feature extraction value from the normalization results

Contrast, here is a contrast calculation using
$Contrast$ (3.1) = (3 − 1)2 × 0.167 = 0.668
$Contrast$ (4.3) = (4 − 3)2 × 0.333 = 0.333
$Contrast$ (6.4) = (6 − 4)2 × 0.333 = 1.332
$Contrast$ (7.6) = (7 − 6)2 × 0.167 = 0.167
$Contrast$ = 0.668 + 0.333 + 1.332 + 0.167 = 2.333

Dissimilarity, below is the calculation of dissimilarity
$Dissimilarity$ (3,1) = |3 − 1| × 0.167 = 0.333
$Dissimilarity$ (4,3) = |4 − 3| × 0.333 = 0.333
$Dissimilarity$ (6,4) = |6 − 4| × 0.333 = 0.667
$Dissimilarity$ (7.6) = |7 − 6| × 0.167 = 0.167
$Dissimilarity$ = 0.333 + 0.333 + 0.667 + 0.167 = 1.5

Homogeneity, the following is a calculation of homogeneity
$Homogenity$ (3.1) =0.167/ 〖 1+(3-1) 〗 ^2 = 0.033
$Homogenity$ (4.3) =0.333/ 〖 1+(4-3) 〗 ^2 = 0.167
$Homogenity$ (6.4) = 0.333/ 〖 1+(6-4) 〗 ^2 = 0.067
$Homogenity$ (7.6) = 0.167/ 〖 1+(7-6) 〗 ^2 = 0.084

Energy, the following is an energy calculation
$Energy$ = 0.1672 + 0.3332 + 0.3332 + 0.1672 = 0.278

Correlation, below is the Correlation calculation
μi = (3 × 0.167) + (4 × 0.333) + (6 × 0.333) + (7 × 0.167) = 5
μj = (1 × 0.167) + (3 × 0.333) + (4 × 0.333) + (6 × 0.167) = 3.5

σi (3.1) = (3 − 5)2 × 0.167 = 0.668
σi (4.3) = (4 − 5)2 × 0.333 = 0.333
σi (6.4) = (6 − 5)2 × 0.333 = 0.333
σi (7.6) = (7 − 5)2 × 0.167 = 0.668
σi = 0.668 + 0.333 + 0.333 + 0.668 = 2.002
σj (3.1) = (1 − 3.5)2 × 0.167 = 1.044
σj (4.3) = (3 − 5)2 × 0.333 = 0.083
σj (6.4) = (4 − 5)2 × 0.333 = 0.083
σj (7.6) = (6 − 5)2 × 0.167 = 1.044
σj = 0.668 + 0.333 + 0.333 + 0.668 = 2.254

$Correlation$ (3.1) = ((3-5) x (1-3.5) x 0.167) / (0.668 x 1.044) = 1.198
$Correlation$ (4.3) = ((4-5) x (1-3.5) x 0.333) / (0.333 x 0.083) = 6.006
$Correlation$ (6.4) = ((6-5) x (4-3.5) x 0.333) / (0.333 x 0.083) = 6.006
$Correlation$ (7.6) = ((3-5) x (1-3.5) x 0.167) / (0.668 x 1.044) = 1.198
$Correlation$ = 1.198 + 6.006 + 6.006 + 1.198 = 14.407

The following is data from several feature extractions that have been combined into one data.

Table 1. *Example of feature extraction data*

Mean-R	Mean-G	Mean-B	Mean-H	Mean-V	Mean-s	Mean-Gray	Standar-Deviasi	Luas	Keliling	Contrast	Dissimilarity	Homogeneity	Energy	Correlation	enis Kelapa
28.72951	20.71148509	8.633694	3.820883469	28.73336	34.72736	21.73255014	48.91498583	33341	2567.042	77.10647	2.892961455	0.81277291	0.783587	0.98390801	0
24.07994	16.9786716	7.078736	2.903175309	24.08172	28.21663	17.97422469	45.6782213	25307	4453.477	58.47973	2.276562809	0.849830261	0.827743	0.986003713	0
34.21065	24.21836516	10.97447	3.76576929	34.21218	36.57387	25.69879297	52.43658203	33204	3829.237	38.70116	2.126966636	0.803857881	0.767803	0.992971382	0
83.9593	66.81858239	36.80381	10.23165126	83.96065	74.48853	68.50519987	71.72839069	55227.5	2608.373	92.5297	5.104272732	0.517922718	0.429047	0.991009549	0
80.33883	57.11953408	29.74632	8.648356896	80.34007	82.32262	60.95604559	63.21773142	25645	2350.993	71.47866	4.56708829	0.523586976	0.431036	0.991058753	0
42.11324	32.04624183	19.92694	4.655812636	42.1182	34.74939	33.67236819	61.8925967	75630.5	3919.656	32.89976	2.084533159	0.763619894	0.67117	0.995709722	0
54.22509	34.50275587	19.97983	5.486919657	54.2276	64.22748	38.76149119	51.39509521	7942	2363.503	94.95732	4.608974359	0.619340287	0.54905	0.982041769	0
21.8668	14.40963387	12.97617	12.0352599	21.93709	25.4012	16.47922357	35.0309865	1296	646.725	65.532	2.773286845	0.791223556	0.754808	0.97333363	1
12.47958	7.351338624	6.768484	9.208867725	12.51494	24.49738	8.818886243	22.50935936	642	217.8234	30.19734	1.593513514	0.841240113	0.808263	0.970240315	1
15.32239	9.197689815	8.199803	10.701125	15.3779	25.2045	10.9159537	24.02225726	40.5	28.72792	6.110241	0.792647195	0.845143594	0.776642	0.99471163	1
38.59588	24.38083932	21.3782	26.27008662	38.77021	49.3132	28.29613741	37.9399178	845	510.3574	87.68699	4.519184888	0.587086637	0.49901	0.969575925	1
32.68512	21.07375658	18.18218	17.60440157	32.81044	40.97558	24.22296658	41.58667762	2744.5	1223.894	77.16403	3.483828407	0.693636412	0.6271	0.977721385	1
68.25299	44.34785098	21.76282	6.453411749	68.2535	72.41065	48.92796778	63.09698793	50446	3913.06	136.652	5.717123729	0.582337528	0.517587	0.982853048	1
55.33452	33.46037138	30.44818	36.59985335	55.68197	80.81039	39.66488742	38.99239547	684.5	218.5513	107.844	5.843722944	0.435583845	0.335189	0.964529884	1

Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (143-156)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.41225

3.4. Split Dependent dan Independent Value

After the value from feature extraction is obtained, the next step is to separate the data into dependent and independent data. Dependent value, also known as the target variable, contains variables from colour feature extraction, producing the attributes meanR, meanG, meanB, meanH, meanS, meanV, mean grayscale and mean standard deviation. Size feature extraction will produce area and perimeter attributes. Meanwhile, for Gray Level feature extraction

Co-occurrence Matrix (GLCM) will produce contrast, dissimilarity, homogeneity, energy and correlation attributes. Then, the Independent Value is the quality of the coconut, which is Standard coconut and Non-Standard Coconut. To carry out the separation, the independent value will be entered into variable x, and the Dependent Value will be entered into variable y.

3.5. Scaling Features

The feature scaling that will be used is the Normalization Method (Min-max Scaling) where each feature will be changed so that it is within a certain range such as [0,1]. Meanwhile, the data used for Feature Scaling is dependent data which has very different ranges. The results of Feature Scaling can be seen in the following table:

Table 2. *Feature scaling result*

Mean-R	Mean-G	Mean-B	Mean-H	Mean-V	Mean-s	Mean-Gray	ndar-Devi	Luas	Keliling	Contrast	Dissimilarity	Homogeneity	Energy	Correlation
0.158503331	0.162721572	0.077789	0.045244	0.158249	0.111975	0.160367	0.438772	0.29904	0.276639	0.231377	0.229237957	0.895597716	0.905427	0.849103168
0.113150967	0.117257304	0.044376	0.018514	0.112861	0.056533	0.113695	0.384988	0.226972	0.481132	0.170672	0.161961374	0.972669911	0.978981	0.87311118
0.211966809	0.205434055	0.128088	0.043639	0.211708	0.127699	0.209622	0.497289	0.297811	0.413463	0.106214	0.14563376	0.877056163	0.879134	0.952931614
0.69721958	0.724288718	0.683111	0.231971	0.697121	0.450562	0.74121	0.817854	0.495369	0.281119	0.281641	0.470590639	0.282365772	0.314841	0.930457184
0.626481838	0.572157036	0.540754	0.203528	0.626565	0.475607	0.614241	0.670568	0.497064	0.520266	0.462386	0.609951411	0.269286197	0.317345	0.819019884
0.267789566	0.274602021	0.259428	0.061207	0.267558	0.112055	0.280929	0.592759	0.629381	0.390684	0.171936	0.225929585	0.799435924	0.790391	0.933950051
0.661905214	0.606157949	0.531459	0.185854	0.661794	0.517273	0.647461	0.676435	0.230004	0.253219	0.213036	0.411959859	0.294146344	0.318155	0.931020853
0.289049363	0.300774656	0.320459	0.069563	0.28885	0.112163	0.308641	0.654416	0.678391	0.423265	0.087307	0.141002375	0.793368856	0.718165	0.984301559
0.407189427	0.330694075	0.321596	0.093771	0.407005	0.363184	0.37184	0.479983	0.071202	0.254575	0.289552	0.416531497	0.493294898	0.51474	0.82772383
0.814033691	0.799457506	0.718391	0.243263	0.813979	0.560719	0.833092	0.789468	0.374763	0.411164	0.322761	0.549893549	0.175484726	0.236817	0.910548133
0.769976453	0.708633353	0.707551	0.206475	0.769973	0.508263	0.761376	0.755089	0.610297	0.519601	0.187936	0.407350785	0.226433739	0.243419	0.9543231
0.396572074	0.385721174	0.279961	0.111957	0.396381	0.260405	0.397735	0.696105	0.556462	0.342661	0.215798	0.304625406	0.647387279	0.660808	0.933663166
0.351515836	0.381282103	0.311229	0.133019	0.351368	0.208544	0.379909	0.729082	0.592024	0.366895	0.195375	0.255882386	0.709825817	0.697716	0.947670005
0.674910507	0.631150464	0.53921	0.189188	0.674801	0.49666	0.667815	0.733014	0.448867	0.424788	0.247961	0.42904525	0.326638663	0.354649	0.927409312

3.6. Splitting Data (Testing and Training)

At this stage, the dataset is divided into two parts, namely training data and testing data. The division between training data and testing data will be carried out in several scenarios, namely 50:50, 60:40, 70:30, 80:20 and 90:10, after which splitting data will be selected with the highest accuracy value.

3.7. K-Nearest Neighbour Algorithm Classification

The author creates a model using the K-Nearest Neighbour Algorithm at this stage. The K-nearest neighbour algorithm is used to classify Images by comparing testing data and training data using the Euclidean distance formula with several K-value scenarios. The author uses the library from sci-kit-learn to create a KNN machine learning model.

```
from sklearn.neighbors import KNeighborsClassifier
knn= KNeighborsClassifier(n_neighbors= 3 ,metric='euclidean').fit(x_train,y_train)
knn
```

```
KNeighborsClassifier
KNeighborsClassifier(metric='euclidean')
```

Figure 13. *Scripting to create a K-NN model with the scikit-learn library*

Based on Image 13, the K value is 3, while the metric uses Euclidean so that Image classification is carried out by comparing training data and testing data using the Euclidean formula.

a. K-NN Model Testing Scenario

The first step is to test all K values from 1-10 on specific data splitting. From the results obtained for the combination of K values and splitting data after several experiments, the following values were obtained:

Table 3. *Scenario K-NN results with splitting data and number of K*

Splitting Data	K=1	K=3	K=5	K=7	K=9
90:10	86	90	96	88	90
80:20	86	90	93	90	92
70:30	90.67	92	90.67	90	90.67
60:40	91	89.5	88.5	90	89.5
50:50	90.4	90.8	90	91.6	91.6

Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (143-156)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.41225

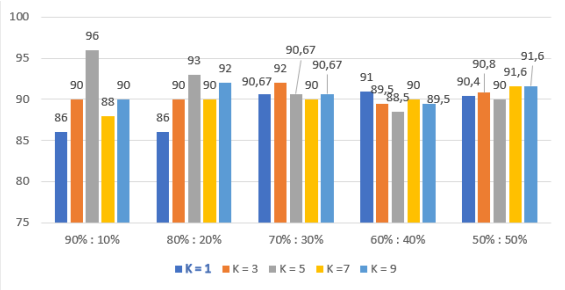

Figure 15. *Graph of K-NN scenario results with splitting data and number of K*

Table 3 and Figure 15 show that the highest accuracy value in Splitting data is 90:10 with a value of K = 5, so the model has been thoroughly evaluated using the Confusion Matrix and will be developed for the Deployment stage.

3.8. Comparison K-NN with SVM, Random Forest and CNN Algorithm Models

The next step, after selecting the best model from K-Nearest Neighbors (K-NN) with a data split of 90:10 and K = 5, is to compare this model with three others: Support Vector Machine (SVM), Random Forest, and Convolutional Neural Network (CNN).

For the SVM, both the linear kernel and Radial Basis Function (RBF) kernel will be utilized. The cost of misclassification, or the regularization parameter, will be set to 1 (C = 1). The Random Forest model will use 100 trees (n_estimators = 100, random_state = 0). For the CNN, the "Adam" optimizer will be employed, with 20 epochs and a batch size of 32.

3.9. Evaluation

The following process is the evaluation stage where, in this case, the model that has the best accuracy, namely the model with data splitting 90:10 and a value of k = 5, will be thoroughly evaluated using a confusion matrix so that the values for accuracy, precision, recall, F-1 Score can be obtained.

3.10. Accuracy

The following is the accuracy value of the model with data splitting 90:10 and k = 5

Accuracy = 48 / 50 x 100% = 96%

In data testing using accuracy metrics, an accuracy of 96% was obtained. Apart from getting accuracy values, namely to get precision, recall and F-1 Score values, the author uses the Confusion Matrix. Moreover, as a result of the Confusion Matrix, the following results are obtained:

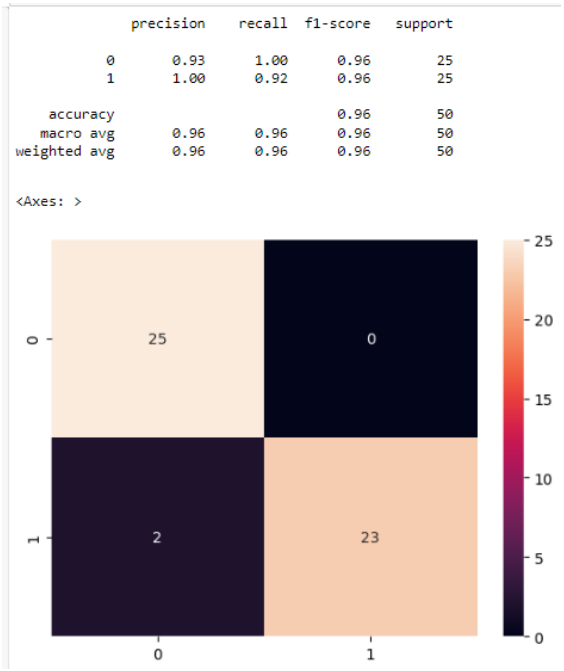

Figure. *Confusion matrix results*

Based on Figure 16, the 2×2 confusion matrix consists of two parts, namely predicted results and actual results. The actual results consist of two classes, namely classes 0 and 1. Where 0 represents the Standard Coconut type, 1 represents the Non-Standard Coconut. Meanwhile, the prediction results also consist of classes 0 and 1.

In the case of the Standard Coconut class, we have a True Positive (TP) value of 25. The True Negative (TN) value is 23. The False Positive (FP) value is 2. The False Negative (FN) value is 0.

Meanwhile, we have a True Positive (TP) of 23 for the non-standard coconut class. A True Negative (TN) value of 25. A False Positive (FP) value of 0. A False Negative (FN) value of 2.

1. Precision can be calculated

   Precision (Standard Coconut) = 25 / (25+2) = 0.93

   Precision (Nonstandard Coconut) = 23 / (23+0) = 1

The precision value is the percentage of positive cases identified correctly from all cases identified as positive by the model. The precision value obtained for non-standard coconut was 93%, and non-standard coconut

Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (143-156)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.41225

was 100%. The highest precision value was obtained by non-standard coconut.

2. Recall can be calculated using the following calculation:

   Recall (Standard Coconut) = 25/(25+0)=1

   Recall (Nonstandard Coconut) = 23/(23+2)=0.92

The recall value is the percentage of positive cases correctly identified by the model out of all positive cases. The recall value obtained for standard coconut was 100%, and for non-standard coconut, it was 92%. The highest recall value was obtained by standard coconut.

3. The F-1 score can be calculated using the following formula.

   F-1 Score (Standard Coconut) = (2 x 1 x 0.93) / (1+0.93) = 0.96

   F-1 Score (Standard Coconut) = (2 x 0.92 x 1) / (0.92 +1) = 0.96

F1-score is the harmonic average of precision and recall for each class. The F1-score value obtained for standard coconut was 96%, and non-standard coconut was 96%. So, the F1-score value for standard and non-standard coconut has the same value.

3.11. Comparison with KNN Evaluation Results with SVM, Random Forest and CNN Algorithm Models

Table 4. *Comparison with KNN evaluation results with SVM, random forest and CNN algorithm models*

Algorithm	accuracy	precision	recall	F1-score
K-NN	0.96	0.96	0.96	0.96
SVM (Kernel Linear)	0.9	0.91	0.9	0.9
SVM (Kernel RBF)	0.94	0.94	0.94	0.94
Random Forest	0.92	0.92	0.92	0.92
CNN	0.86	0.87	0.86	0.86

The table 4 compares the performance of five classification algorithms based on four main evaluation metrics: accuracy, precision, recall, and F1-score. The K-Nearest Neighbor (K-NN) algorithm shows the best performance with a perfect score of 0.96 on all metrics, indicating high accuracy and consistency of predictions. SVM with RBF kernel follows with a score of 0.94 evenly, indicating its ability to handle non-linear data well. Random Forest is in third place with a score of 0.92 for all metrics, indicating its stability and effectiveness as an ensemble model. SVM with Linear kernel has a slightly lower score, with a precision reaching 0.91, but other metrics remain at 0.90, reflecting limitations on non-linear data. Although based on deep learning, CNN shows the lowest performance (accuracy 0.86 and F1-score 0.86), likely due to limited training data or a suboptimal model architecture. This table indicates that classical algorithms such as K-NN and SVM RBF can provide superior classification results compared to deep learning approaches in the data context.

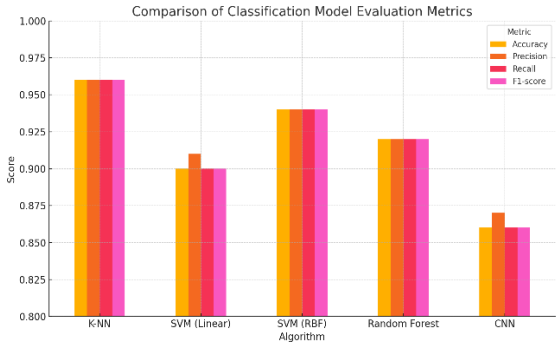

Figure 17. *Comparison chart of KNN evaluation results with svm, random forest and CNN algorithm models*

Image 17 visually compares the performance of five classification algorithms based on the metrics of accuracy, precision, recall, and F1-score. The K-NN model stands out with the highest scores in all four metrics, followed by SVM with RBF kernel and Random Forest. CNN shows the lowest performance, which is likely influenced by the large data requirement and complex tuning parameters. This graph reinforces the results shown in the previous table and emphasizes the importance of selecting the appropriate algorithm based on the dataset's characteristics.

3.12. Deployment

After the model with the best accuracy is obtained, namely the K-NN Algorithm with 90:10 data splitting with a value of k = 5, the next step is to deploy the model into a web-based application prototype. To create a model that can be implemented on the web, the first step is to convert the K-NN model into pickle form. After the model is in pickle form, the next step is to create a web-based application prototype. The prototype was developed using Python, with Visual Studio Code as the IDE and

the Streamlit library. The following are the results of the application that has been built.

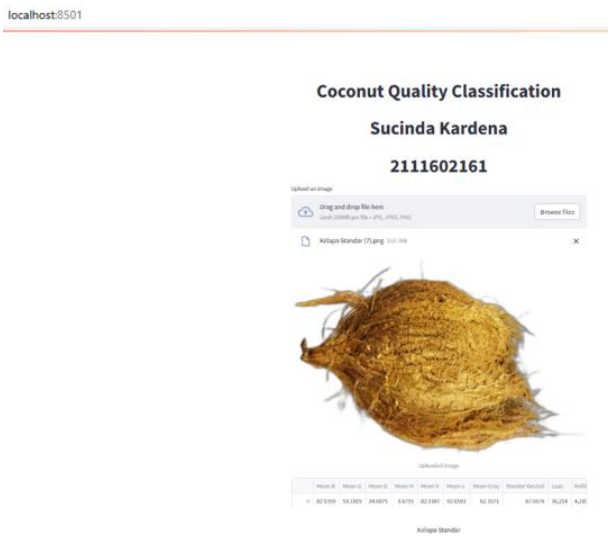

Figure 18. *Coconut classification web application*

CONCLUSION

This study shows that coconut quality can be effectively classified by extracting key features, including colour, texture, and shape. The model that achieved the highest performance was K-NN using a 90:10 training-test data split with k=5k, resulting in 96% accuracy. SVM with RBF kernel was followed with an accuracy score of 94%, indicating its ability to handle non-linear data well. Random Forest came in third with an accuracy score of 92%, indicating its stability and effectiveness as an ensemble model. SVM with Linear kernel had a slightly lower score, with an accuracy value of 90%, and CNN showed the lowest performance, with an accuracy of 86%. Then, from the best model, namely K-NN, we can develop it into a web application that is used for predicting coconut quality classification and can also be embedded into a coconut quality separator machine so that coconuts can be classified automatically through the machine

REFERENCES

[1] R. Sari, "Strategi Pemerintah Kabupaten Indragiri Hilir Dalam Menjaga Stabiltas Harga Guna Meningkatkan Kesejahteraan Petani Kelapa Di Kecamatan Batang Tuaka," *Jisip-Unja*, vol. 5, no. 1, pp. 52–63, 2021.

[2] H. Anggrasari, A. K. Sari, and F. R. Arminda, "Indonesian Coconut Oil Export Opportunities with Main Trade Partner Countries in the International Market," *Bul. Penelit. Sos. Ekon. Pertan. Fak. Pertan. Univ. Haluoleo*, vol. 25, no. 1, pp. 44–55, 2023, doi: 10.37149/bpsosek.v25i1.445.

[3] R. Nandaputra, H. T. Sukmana, S. Aripiyanto, Y. Durrachman, D. Khairani, and S. U. Masruroh, "Transfer Learning for Coconut Quality Classification Using the Pretrained Model Efficientnet," in *2024 12th International Conference on Cyber and IT Service Management (CITSM)*, 2024, pp. 1–7. doi: 10.1109/CITSM64103.2024.10775588.

[4] T. C. Lim, J. O. Torregosa, A. R. A. Pescadero, and R. S. Pangantihon, "De-husked Coconut Quality Evaluation using Image Processing and Machine Learning Techniques," *ACM Int. Conf. Proceeding Ser.*, pp. 28–33, 2019, doi: 10.1145/3383783.3383789.

[5] S. Siddesha and S. K. Niranjan, "Color and texture in classification of coconut," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 8, pp. 1745–1750, 2019.

[6] N. A. Fadchar and J. C. D. Cruz, "A Non-Destructive Approach of Young Coconut Maturity Detection using Acoustic Vibration and Neural Network," *Proc. - 2020 16th IEEE Int. Colloq. Signal Process. its Appl. CSPA 2020*, no. Cspa, pp. 136–140, 2020, doi: 10.1109/CSPA48992.2020.9068723.

[7] J. A. Widians, H. S. Pakpahan, E. Budiman, H. Haviluddin, and M. Soleha, "Klasifikasi Jenis Bawang Menggunakan Metode K-Nearest Neighbor Berdasarkan Ekstraksi Fitur Bentuk dan Tekstur," *J. Rekayasa Teknol. Inf.*, vol. 3, no. 2, p. 139, 2019, doi: 10.30872/jurti.v3i2.3213.

[8] H. P. Hadi and E. H. Rachmawanto, "Ekstraksi Fitur Warna Dan Glcm Pada Algoritma Knn Untuk Klasifikasi Kematangan Rambutan," *J. Inform. Polinema*, vol. 8, no. 3, pp. 63–68, 2022, doi: 10.33795/jip.v8i3.949.

[9] S. F. A. Wijaya, K. Koredianto, and S. Saidah, "Analisis Perbandingan K-Nearest Neighbor dan Support Vector Machine pada Klasifikasi Jenis Sapi dengan Metode Gray Level Coocurrence

Matrix," *J. Ilmu Komput. dan Inform.*, vol. 2, no. 2, pp. 93–102, 2022, doi: 10.54082/jiki.27.

[10] M. Muchtar and R. A. Muchtar, "Perbandingan Metode Knn Dan Svm Dalam Klasifikasi Kematangan Buah Mangga Berdasarkan Citra Hsv Dan Fitur Statistik," *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 2, pp. 876–884, 2024, doi: 10.23960/jitet.v12i2.4010.

[11] E. F. Vermote, S. Skakun, I. Becker-Reshef, and K. Saito, "Remote sensing of coconut trees in Tonga using very high spatial resolution WorldView-3 data," *Remote Sens.*, vol. 12, no. 19, pp. 1–8, 2020, doi: 10.3390/RS12193113.

[12] Y. Prabowo and K. N. Nasahara, "Detecting and Counting Coconut Trees in Pleiades Satellite Imagery Using Histogram of Oriented Gradients and Support Vector Machine," *Int. J. Remote Sens. Earth Sci.*, vol. 16, no. 1, p. 87, 2019, doi: 10.30536/j.ijreses.2019.v16.a3089.

[13] L. M. B. Alonzo, F. B. Chioson, H. S. Co, N. T. Bugtai, and R. G. Baldovino, "A machine learning approach for coconut sugar quality assessment and prediction," *2018 IEEE 10th Int. Conf. Humanoid, Nanotechnology, Inf. Technol. Commun. Control. Environ. Manag. HNICEM 2018*, pp. 1–4, 2019, doi: 10.1109/HNICEM.2018.8666315.

[14] A. Aquino, M. G. A. Bautista, A. Bandala, and E. Dadios, "Color quality assessment of coconut sugar using Artificial Neural Network (ANN)," *8th Int. Conf. Humanoid, Nanotechnology, Inf. Technol. Commun. Control. Environ. Manag. HNICEM 2015*, no. December, 2016, doi: 10.1109/HNICEM.2015.7393182.

[15] D. Nesarajan, L. Kunalan, M. Logeswaran, S. Kasthuriarachchi, and D. Lungalage, "Coconut Disease Prediction System Using Image Processing and Deep Learning Techniques," *4th Int. Conf. Image Process. Appl. Syst. IPAS 2020*, pp. 212–217, 2020, doi: 10.1109/IPAS50080.2020.9334934.

[16] I. Bhat, U. V, N. Jagadeesh, S. Bhat, and R. S. Shenoy, "Tender Coconut Classification using Decision Tree and Deep Learning Technique," *2023 10th Int. Conf. Signal Process. Integr. Networks*, pp. 395–398, 2023, doi: 10.1109/SPIN57001.2023.10117353.

[17] A. S. Sagayaraj, T. K. Devi, and S. Umadevi, "Prediction of Sulfur Content in Copra Using Machine Learning Algorithm," *Appl. Artif. Intell.*, vol. 35, no. 15, pp. 2228–2245, 2021, doi: 10.1080/08839514.2021.1997214.

[18] J. A. Caladcad *et al.*, "Determining Philippine Coconut Maturity Level Using Machine Learning Algorithms Based On Acoustic Signal," *Comput. Electron. Agric.*, vol. 172, no. November 2019, p. 105327, 2020, doi: 10.1016/j.compag.2020.105327.

[19] K. Vishruth, G. B. Srujana, and S. Shetty, "Analyzation of Quality of Coconut," no. May, pp. 5522–5527, 2019.

[20] H. T. Sukmana, Puspitasari, A. Alamsyah, S. Aripiyanto, and L. K. Oh, "Improving Performance of Copra Type Classification Using Feature Extraction With K-Nearest Neighbour Algorithm," *Int. J. Ebus. eGovernment Stud.*, vol. 15, no. 1, pp. 512–532, 2023, doi: 10.34111/ijebeg.2023150123.

[21] A. Anton, N. F. Nissa, A. Janiati, N. Cahya, and P. Astuti, "Application of Deep Learning Using Convolutional Neural Network (CNN) Method For Women's Skin Classification," *Sci. J. Informatics*, vol. 8, no. 1, pp. 144–153, 2021, doi: 10.15294/sji.v8i1.26888.

[22] H. Mayatopani, R. I. Borman, W. T. Atmojo, and Arisantoso, "Classification of Vehicle Types Using Backpropagation," *J. Ris. Inform.*, vol. 4, no. 1, 2021, [Online]. Available: https://ejournal.kresnamediapublisher.com/index.php/jri/article/view/139

[23] Suharjito, B. Imran, and A. S. Girsang, "Family relationship identification by using extract feature of gray level co-zoccurrence matrix (GLCM) based on parents and children fingerprint," *Int. J. Electr. Comput. Eng.*, vol. 7, no. 5, pp. 2738–2745, 2017, doi: 10.11591/ijece.v7i5.pp2738-2745.

[24] D. Rajput, W. J. Wang, and C. C. Chen, "Evaluation of a decided sample size in machine learning applications," *BMC Bioinformatics*, vol. 24, no. 1, pp. 1–17,

2023, doi: 10.1186/s12859-023-05156-9.
[25] "ICAO document 07 - Methodology for Actual Life Cycle Emissions - March 2024.pdf," no. March, 2024.
[26] E. A. B. Flores, H. V. Olivera, I. C. M. Valencia, and C. F. M. Cubas, "Fruit Fly Classification (Diptera: Tephritidae) in Images, Applying Transfer Learning," 2025, [Online]. Available: http://arxiv.org/abs/2502.00939