OPEN ACCESS
VOL. 6, NO. 1, PP.031-041, NOVEMBER 2023

ISSN : 2622-8106 (ONLINE)
DOI: 10.20895/INISTA.V6I1.1273

Journal of Informatics, Information System, Software Engineering and Applications (INISTA)

Classification of Oil Palm Ripeness Level Using Learning Vector Quantization (LVQ) Method

Deddy Rudhistiar [1], Widhy Wahyani [2], Thesa Adi Saputra Yusri *[3]

[1] Program Studi Teknik Informatika, Institut Teknologi Nasional Malang
JL. Raya Karanglo KM. 2, Tasikmadu, Kec. Lowokwaru, Kota Malang, Jawa Timur 65153, Indonesia.
[1] rudhistiar@lecturer.itn.ac.id

[2] Program Studi Bisnis Digital, Institut Teknologi Nasional Malang
JL. Raya Karanglo KM. 2, Tasikmadu, Kec. Lowokwaru, Kota Malang, Jawa Timur 65153, Indonesia.
[2] widhy_wahyani@lecturer.itn.ac.id

[3] Program Studi Matematika, Universitas Negeri Yogyakarta
Jl. Colombo No.1, Karang Malang, Caturtunggal, Sleman, Daerah Istimewa Yogyakarta, Indonesia
[3,*] thesaadisaputrayusri@uny.ac.id

Received on 20-09-2023, revised on 06-12-2023, accepted on 6-12-2023

Abstract

The palms fruit plantation industry is the most vital section industry in Indonesia. This is need to be categorized the essential for farmers to determine the appropriate harvest time, leading to increased palm fruit production and quality. One available classification method is the Learning Vector Quantization (LVQ) method, which is a type of artificial neural network. of artificial neural network commonly used for tasks involving pattern recognition and classification. The implementation of Learning Vector Quantization (LVQ) to classify the ripeness of oil palm fruits is investigated in this study. In addition, this study provides a comprehensive set of procedures ranging from data collection and pre-processing to training and testing of the LVQ model, and finally, the proposed method has been validated by testing it on previously unseen data. Three feature extraction methods, specifically Gray-Level Co-occurrence Matrix (GLCM), Hue, Saturation, and Value (HSV), and t-Distributed Stochastic Neighbor Embedding (t-SNE), were assessed for their performance. The results show that the chosen feature extraction method strongly influences the classification performance. The accuracy of the model employing t-SNE features is notably the highest at 50%, indicating its efficacy in identifying the ripeness level of palm fruits by extracting pertinent features. On the other hand, the GLCM feature has a 40% accuracy in the test data, suggesting that although it captures texture information, it may not comprehensively encapsulate ripeness characteristics. Additionally, the HSV feature achieves an accuracy of 45%, which is less precise than that of t-SNE. To conclude, this study elucidates the appropriateness of various feature extraction techniques in categorizing the degree of ripeness in palm fruits. The t-SNE feature extraction model stands out as the most efficient option, exhibiting greater precision in comparison to other methodologies.

Keywords: palm fruit, ripeness, LVQ, GLCM, HSV, t-SNE

Corresponding Author:
*Thesa Adi Saputra Yusri
Program Studi Matematika, Universitas Negeri Yogyakarta,
Jl. Colombo No.1, Karang Malang, Caturtunggal, Sleman, Daerah Istimewa Yogyakarta, Indonesia
thesaadisaputrayusri@uny.ac.id

I. INTRODUCTION

The palms fruit plantation industry holds immense significance as one of Indonesia's vital sectors. Accurate categorization of palm fruit is essential for farmers to determine the appropriate harvest time, leading to increased palm fruit production and quality [1]. The palm fruit plantation industry is a crucial sector in Indonesia. Ripe palm fruit yields outstanding quality produce, underscoring the significance of the classification of palm fruit ripeness [2].

One available classification method is the Learning Vector Quantization (LVQ) method, which is a type of artificial neural network. of artificial neural network commonly used for tasks involving pattern recognition and classification. It is an algorithm which is supervised and uses a set of training data with labels. The LVQ method has applications in different fields, such as image recognition, speech recognition and bioinformatics. This technique can produce precise and efficient classification models in a relatively brief period. LVQ models are extensively employed to categorize the degree of ripeness of fruits such as tomatoes, mangoes, coffee, limes, and other miscellaneous fruits [3]–[7].

Several previous studies have been conducted in the field of palm fruit ripeness classification. One study about used the K-Nearest Neighbor (KNN) classification method to classify palm fruit ripeness, proposed that the KNN method can be used to classify palm fruit ripeness with high accuracy. However, the LVQ method has advantages in time efficiency and higher accuracy compared to the KNN method. In addition, there are also other classification methods such as Support Vector Machine (SVM) [8] and Artificial Neural Networks methods that can be used for palm fruit ripeness classification. However, LVQ method was chosen because it has advantages in time efficiency and higher accuracy compared to other methods [9].

Research related to image processing cannot be separated from the importance of feature engineering, one of which is the feature extraction process. Various techniques and methods of feature extraction in images can be implemented in the classification process. Research about the uses of Gray Level Co-Occurrence (GLCM) features combined with K-Nearest Neighbor to classify leaf images. [10]. Also, the used of GLCM features to classify tangerine fruit (Citrus reticulata Blanco) [11]. Another research about the used HSV features as features combined as input to the deep learning model [12]. So, in this research, we will develop a ripeness classification model of palm fruit using the LVQ method. This research aims to develop an accurate and efficient classification model in classifying the ripeness of palm fruit. In this research, we will process palm fruit image data and classify the ripeness of palm fruit using the LVQ method.

II. RESEARCH METHOD

The aim of this study was to investigate and develop an efficient method to determine the ripeness status of palm fruit using the Learning Vector Quantization (LVQ) technique. Palm fruit was categorized into distinct ripeness levels employing the LVQ approach as a tool. The LVQ method is one of the machine learning methodologies that exhibits the ability to perceive patterns within data and place data in relevant groups. LVQ will be used in this study to comprehend the visual features of palm fruit at different ripeness levels, enabling their high-precision classification. To perform this research, visual data of palm fruit at various ripeness levels was collected. The collected data underwent subsequent processing, followed by LVQ model training and testing against previously unseen data. The methodology's outline is presented in Figure 1.

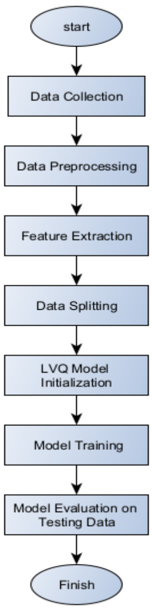


Fig. 1. Research Methodology

A. Data Collection

The initial phase of this research involves the systematic collection of raw data, primarily comprised of images or measurements of palm fruit samples obtained from various sources, such as palm plantations or experimental setups. These data are critical as they constitute the foundation for the subsequent stages of the study. It is essential to ensure that the collected data encompass a diverse representation of palm fruit at different ripeness levels.

B. Data Annotation

Data annotation is a pivotal step in the research process, where domain experts or trained annotators assign categorical labels to each data instance based on its corresponding ripeness level. This manual labeling process is labor-intensive and requires expertise in palm ripeness assessment. Labels may encompass various categories, such as "ripe," "unripe," or specific ripeness stages, enabling supervised learning.

C. Data Preprocessing

Subsequent to data annotation, the collected dataset undergoes meticulous preprocessing. This phase encompasses several essential tasks:

1. Data Cleaning: Removal of any erroneous or incomplete data instances, ensuring data quality and consistency.
2. Normalization: Scaling the data to a common range (e.g., [0, 1]) to facilitate model convergence.
3. Feature Extraction: Extraction of relevant features from the images, such as color histograms, texture descriptors, or shape characteristics, to provide meaningful information to the LVQ model. the performance of three feature extraction techniques, namely Gray-Level Co-occurrence Matrix (GLCM), HSV (Hue, Saturation, Value) and t-Distributed Stochastic Neighbor Embedding (t-SNE), will be tested.
   a. Gray-Level Co-occurrence Matrix (GLCM)

   GLCM is an important tool in image analysis for extracting texture features from grey-scale images. Abbreviations are explained when first used. It measures the frequency of different pairs of pixel intensities appearing together in an image to depict the spatial relationships between pixel intensities. This generates a symmetrical matrix that measures the likelihood of specific intensity pairs occurring at different directions and distances. The text adheres to conventional academic structure and employs clear, objective language with consistent technical terms. It is free from grammatical errors, spelling mistakes, and punctuation errors. Various texture features such as energy, contrast, homogeneity, and correlation can be calculated using GLCM [13].

   Let I be a grayscale image with pixel intensity $I(x, y)$ at coordinates (x, y). The GLCM P of size $N \times N$ will calculate the probability of the occurrence of intensity pairs (i, j) in image I. The mathematical formulation of GLCM shown in Equation 1 [13].

$$P\left(i, j, \delta_x, \delta_y\right) = \frac{1}{N(N-1)} \sum_{x=1}^{N} \sum_{y=1}^{N} I(x, y).I(x+\delta_x, y+\delta_y) \quad (1)$$

   Here, δ_x and δ_y are direction offsets (usually 0, 1, 2, or 3 for horizontal, vertical, and diagonal directions). $P\left(i, j, \delta_x, \delta_y\right)$ is the probability of the occurrence of intensity pair (i, j) in the direction (δ_x, δ_y). Texture features can then be computed from this GLCM matrix.

   b. HSV (Hue, Saturation, Value)

   The HSV colour model represents colour in images and consists of three primary elements: Hue (H), Saturation (S), and Value (V). Hue indicates the actual colour (e.g. red, green, blue), Saturation indicates the brightness level of the colour, and Value indicates the brightness or intensity of the colour. These components can be mathematically expressed in Equation 2 [14].

$$H \in [0,360]; S \in [0,1]; V \in [0,1] \quad (2)$$

c. t-SNE (t-Distributed Stochastic Neighbor Embedding)

t-SNE is a machine learning algorithm utilized in the visualization of high-dimensional data in lower dimensions (typically 2D or 3D). It projects data into a lower-dimensional space whilst preserving structural information between data points. Mathematically, t-SNE functions by evaluating the similarity between data points in the high-dimensional space, aiming to uphold the distances between similar points in the lower-dimensional space. The t-Distributed Stochastic Neighbor Embedding distribution function is employed as the objective function in t-SNE, with its corresponding mathematical formula in Equation 3 [15].

$$C = \sum_i KL(P_i|Q_i) = \sum_i \sum_j p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right) \quad (3)$$

Here, p_{ij} is the conditional probability that point i would choose point j as a neighbor in the high-dimensional space, and q_{ij} is the same probability in the lower-dimensional space. The goal is to minimize this function C to obtain a good projection in the lower-dimensional space.

D. Data Splitting (Training and Testing Data)

The pre-processed dataset is divided into two separate subsets in order to assess the effectiveness of the LVQ model. The Training Dataset is used to train the LVQ model and typically accounts for 70-80% of the data. The training dataset is used to train the LVQ model, which is the recursive process by which the model learns about the underlying patterns and relationships in the data. The remaining 20-30% of the data is deployed as the Testing Dataset. It remains unseen by the model during training and is used solely for performance evaluation.

E. LVQ Model Initialization

Before commencing the training phase, the LVQ model is initialized with specific parameters. These parameters include:

1. Number of Neurons: The number of neurons in the LVQ model, which is typically determined based on the complexity of the problem and the dataset size.
2. Learning Rate: A critical hyperparameter that dictates the magnitude of weight updates during training.
3. Convergence Criteria: Conditions that determine when the training process should stop, such as a predefined number of epochs or reaching a certain accuracy threshold.

General LVQ architecture shown in Figure 2

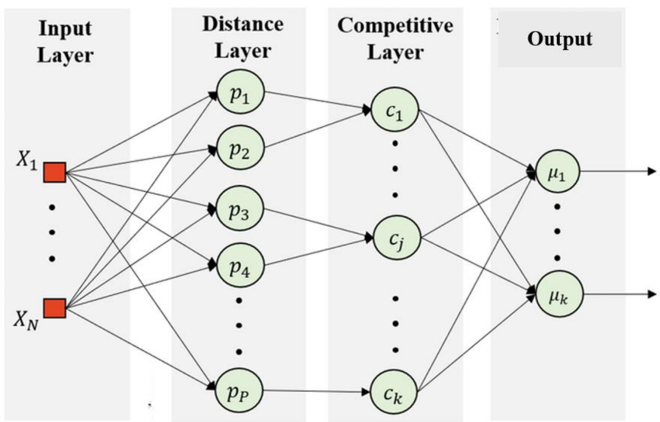


Fig. 2. LVQ Architecture [16]

F. Model Training

This phase is the core of the study, during which the LVQ model undergoes a training process implemented on the given dataset. The training process entails iteratively modifying neuron weights to reduce errors between the model's predictions and the factual ripeness labels. The LVQ model acquires the ability to distinguish between various ripeness levels by relying on the features extracted from the data.

G. Model Evaluation on Testing Data

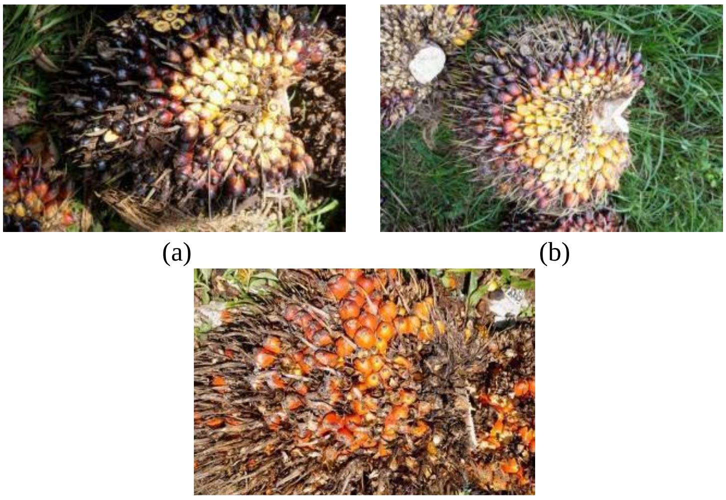
After the training of the model, a comprehensive evaluation is performed on the isolated test set. The effectiveness of the model in ripeness detection is measured through various evaluation metrics such as accuracy, precision, recall, F1-score, and detailed confusion matrices. This stage demonstrates the robustness of the model and allows it to perform well on unseen data.

III. RESULTS AND DISCUSSION

A. Data Preparation

1. Data Collection

Data is collected using secondary sources only. This specific study has gathered data through previous research endeavors. The dataset includes 99 images samples, which are categorized into three classifications; 33 samples portray raw palm, another 33 represent ripe palm, and the remaining images depict overripe palm. These images form the fundamental basis for subsequent analysis and modelling. Every image contains information regarding the state of the palm fruit, whether raw, ripe, or overripe. Several datasets are presented in Figure 3 as examples.



(a) (b) (c)

Fig. 3. (a) Raw Palm (b) Ripe palm (c) Overripe palm

2. Data Preprocessing

After gathering the images, the data is preprocessed to ensure it is appropriate for analysis and model training. One fundamental preprocessing step is cropping, which is used to separate individual palm fruit bunches from the images. This step is critical as it narrows the analysis to the palm fruit bunches, excluding any other extraneous elements that may be present in the original images. By doing this, the analysis becomes more focused and accurate. The outcome of this preprocessing stage is a series of trimmed pictures, each displaying a solitary palm fruit bunch. These cut-out pictures are subsequently employed as the main data for subsequent analysis.

Fig. 4. Cropping process

3. *Data Splitting*

In order to optimize the development and evaluation of machine learning models, the dataset is divided into two subsets, one for training and one for testing. The categorization is allotted proportionally and at random. The training dataset accounts for approximately 80% of the data allocation, while the remaining 20% is delegated for validation and testing. The training dataset is used to instruct the models of machine learning. During this stage, the models acquire patterns and connections in the data, enabling them to make predictions or categorizations. The testing dataset provides an autonomous means to evaluate the model's performance and appraise how effectively the model adapts to new, unrevealed data. If the model performs effectively on the testing dataset, it indicates that it has acquired valuable patterns from the training data.

B. *Feauture Extraction*

1. *HSV*

The feature extraction process using HSV receives RGB image input which is then converted into HSV layers. Each layer is calculated the average and standard deviation to then be used as input to the LVQ model. In this research case the input image is 300x300 pixels, then returns 6 values namely h_mean, s_mean, v_mean, h_std, s_std, v_std. Figure 5 shows an example of feature extraction results with the HSV method.

```
h_mean  = 0.5993
s_mean  = 0.3013
v_mean  = 0.4240
h_std  = 0.1624
s_std  = 0.1897
v_std  = 0.2877
```

Fig. 5. HSV features result

2. *GLCM*

The feature extraction process with GLCM receives input in the form of an RGB image which is then converted to grayscale. then extracts the values of energy, contrast, homogeneity and correlation for directions $0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}$. with these conditions, 16 features are obtained according to their properties and angular direction. Figure 6 shows an example of feature extraction results with the GLCM method.

```
Energy      :  [0.00827796 0.00746843 0.00819582 0.0075383 ]
contrast    :  [1164.51532887 1610.31592488 1135.07617614 1541.88813324]
homogeneity :  [0.08028786 0.06239513 0.07484046 0.06323089]
correlation :  [0.83247715 0.7683992  0.83682721 0.77824266]
```

Fig. 6. GLCM features result

3. *t-SNE*

Feature extraction with t-SNE is the process of reducing the dimensions of complex data into lower dimensions for visualization purposes. This process results in a simpler representation of the data that reflects the structure or patterns present in the data. The main result of t-SNE feature extraction is a representation of the data in a low-dimensional space. This representation is usually a matrix with two or three columns, which can be represented as coordinates of points in a 2D or 3D plot. Figure 7 shows an example of feature extraction results with the t-SNE method.

```
t_SNE_feature :  [-33.960815  30.883009 -29.054983]
```

Fig. 7. t-SNE features result

C. Model initialization

LVQ model development in this study is built based on the number of inputs and classes defined as outputs. As with the feature extraction process, the model input will adjust to the type of feature used. When the input feature uses HSV, the number of inputs $(X_1, X_2, \ldots, X_N)$ is 6 while for GLCM there are 16 inputs. The basic LVQ model used in this research is the LVQ model developed by Sato and Yamada [17]. The function for quantification is squared-Euclidean. the activation function on the objective function uses a sigmoid function.

D. Evaluation

Evaluation and testing of model performance pay attention to the results of the confusion matrix and its derived metrics such as accuracy, F1-score, sensitivity etc. Table I shown result for each model

TABLE I. CLASSIFICATION METRIC EVALUATION

Feature	**Accuracy**	**Precision**			**Recall**			**F1-score**		
		Class 0	Class 1	Class 2	Class 0	Class 1	Class 2	Class 0	Class 1	Class 2
t-SNE	0.50	0.50	0.62	0.40	0.12	0.83	0.67	0.20	0.71	0.50
HSV	0.45	1.00	0.38	0.40	0.25	0.83	0.33	0.40	0.53	0.36
GLCM	0.40	0.46	0.00	0.29	0.75	0.00	0.33	0.57	0.00	0.31

*Class 0 : raw/unripe; Class 1 : ripe; Class 2 : overripe

1. LVQ model with HSV Feature

The confusion matrix presented in Figure 8 shows the performance of the classification model in a task with three different classes. The first row shows five examples correctly classified as class 0 (true positives), but 19 examples incorrectly classified as class 1 or 2 (false negatives), even though they belong to class 0. The second row achieved 31 correct predictions of class 1 (True Positives), yet 2 examples were incorrectly classified as class 0 or class 2, which should have belonged to class 1 (False Negatives). The third row ought to have 18 examples classified as class 2, but they were erroneously predicted as class 1 (False Negatives), whereas 15 examples were accurately classified as class 2 (True Positives). This confusion matrix offers an understanding of the model's classifying accuracy within the dataset and its ability to predict positives and negatives within each class.

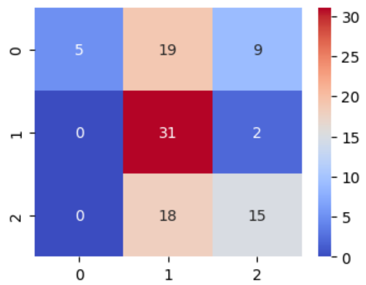


Fig. 8. LVQ model with HSV feature confusion matrix

In this particular context, the model has identified three classes: class 0, class 1, and class 2. Precision is a measure of the model's accurate positive predictions. Class 0 exhibits high precision (1.00), signifying that the model almost perfect in recognizing class 0. Nevertheless, precision for classes 1 and 2 is lower (0.38 and 0.40), indicating that the majority of positive predictions for these two classes are inaccurate. Recall is a metric that gauges how comprehensively the model can identify all the instances that belong to a given class. The high recall rate of Class 1 (0.83) indicates that the model has correctly identified most of the instances of Class 1. However, for Classes 0 and 2, the recall rate (0.25 and 0.33, respectively) is lower, implying that the model found it challenging to recognize a majority of the examples in these two classes. F1-score is a composite measure that combines precision and recall. Class 1 displays a relatively high F1-score of 0.53, suggesting that the model achieves a commendable balance between precision and recall for this category. However, class 0 and class 2 exhibit lower F1-scores of 0.40 and 0.36 respectively that indicating inferior performance in combining precision and recall.

The model's overall accuracy is 0.45, signifying that it accurately predicts merely 45% of the instances in the dataset. While accuracy can give an overview of the model's performance, metrics such as precision, recall, and F1-score provide a more in-depth understanding of how well the model can classify each class individually. Upon analyzing the confusion matrix, it is evident that the model has varying precision and recall for each class, demonstrating diverse performance. Notably, there is a substantial difference in the performance of class 1 compared to class 0 and class 2. Hence, it may be necessary to take corrective actions to enhance the model's accuracy in categorizing classes 0 and 2, while preserving high performance for class 1. The classification report is visually shown in Figure 9.

	precision	recall	f1-score	support
0	1.00	0.25	0.40	8
1	0.38	0.83	0.53	6
2	0.40	0.33	0.36	6
accuracy			0.45	20
macro avg	0.59	0.47	0.43	20
weighted avg	0.64	0.45	0.43	20

Fig. 9. LVQ model with HSV feature classification report

2. LVQ with GLCM Feature

The confusion matrix depicted in Figure 10 exhibits the classification model's performance with GLCM feature extraction. The main diagonal of the matrix exhibits the number of accurately classified examples per class. For instance, along the main diagonal, there are 19 instances from class 0 which were correctly predicted as class 0.

On the other side of the main diagonal there are other numbers, which are the differences between the prediction and the actual class. For example, in the first row, the number 2 represents the number of instances from class 0 that were incorrectly predicted as class 1, whereas the number 12 indicates that 12 instances from class 0 were incorrectly predicted as class 2.

In addition, the matrix shows an uneven distribution of examples across the classes, as indicated by the different number of examples in each row and column. This factor holds significance in assessing model performance, particularly when an imbalance exists amidst the classes.

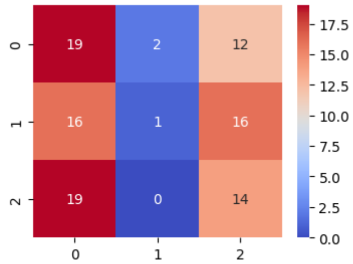


Fig. 10. LVQ model with GLCM feature confusion matrix

The report on the evaluation of the classification model's performance provides valuable insights into its ability to accurately classify the three distinct classes (0, 1, and 2) within the dataset. The precision, recall, and F1-score metrics were utilized to gauge the model's success in this regard. Notably, Class 0 exhibited a favorable precision score (0.46), thereby indicating that the majority of positive predictions for this class were accurate. However, Class 1 exhibits significantly reduced precision at 0.00, indicating a high occurrence of false positive predictions. Class 2 similarly shows lower precision at 0.29. Regarding recall, Class 0 demonstrates high recall at 0.75, implying accurate identification of most Class 0 instances by the model. However, Class 1 has very low recall at 0.00, indicating that the model frequently fails to identify Class 1 examples. Class 2 possesses medium recall at 0.33. The F1-score, which measures the balance between precision and recall, indicates that Class 0 has a relatively good score (0.57), whereas Class 1 has a poor score (0.00) and Class 2 has a low score (0.31). The model's overall accuracy was 0.40, suggesting that it correctly predicted only about 40% of the entire dataset. These findings demonstrate that the model's performance differs between each class, with Class 0 performing better in terms of precision and recall. However, there is a significant need for improvement in predicting class 1. Figure 11 provides a visual representation of the classification report.

	precision	recall	f1-score	support
0	0.46	0.75	0.57	8
1	0.00	0.00	0.00	6
2	0.29	0.33	0.31	6
accuracy			0.40	20
macro avg	0.25	0.36	0.29	20
weighted avg	0.27	0.40	0.32	20

Fig. 11. LVQ model with GLCM feature classification report

3. *LVQ with t-SNE Feature*

Based on Figure 12, the first row of the confusion matrix shows 6 instances of True Positives (TP) for class 0 or raw, 13 instances of False Negatives (FN) for class 1 or ripe, and 14 instances of False Negatives (FN) for class 2 or overripe. In the second row, there are 3 instances of False Negatives (FN) for class 0, 24 instances of True Positives (TP) for class 1, and 6 instances of False Negatives (FN) for class 2. In row three, there are seven instances that fall under False Negatives (FN) for class 0, ten instances that fall under False Negatives (FN) for class 1, and sixteen instances that fall under True Positives (TP) for class 2.

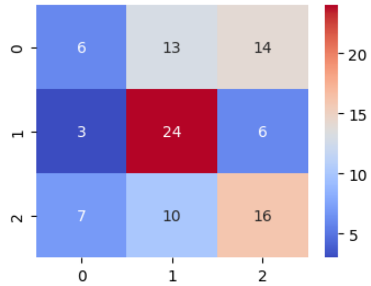


Fig. 12. LVQ model with t-SNE feature confusion matrix

The model's classification performance across three classes (0, 1, and 2) in the dataset is measured using precision, recall, and F1-score metrics as highlighted in figure 12 of the report. Class 1 exhibits the highest precision (0.62), indicating correct positive predictions. Meanwhile, class 0 and 2 exhibit precision values of 0.50 and 0.40, respectively. Class 1 exhibits the highest recall (0.83), indicating accurate identification by the model of most instances pertaining to this class. A recall of 0.67 has been achieved for class 2, while the recall for class 0 is quite low (0.12).

The F1-score balances precision and recall. Class 1 attains the highest F1-score (0.71), indicating a good balance between precision and recall. For class 2, the F1-score is 0.50, while for class 0, it is quite low (0.20). The model's overall accuracy is 0.50, indicating that it can accurately predict only about half of all instances in the dataset.

The report in Figure 13 reveals that the model's performance is varied for each class, with differing levels of precision and recall. There exists a notable gap in performance between class 1 when compared to classes 0 and 2. Consequently, with the aim of enhancing the model's overall performance, improvements must be made in classifying class 0.

	precision	recall	f1-score	support
0	0.50	0.12	0.20	8
1	0.62	0.83	0.71	6
2	0.40	0.67	0.50	6
accuracy			0.50	20
macro avg	0.51	0.54	0.47	20
weighted avg	0.51	0.50	0.44	20

Fig. 13. LVQ model with t-SNE feature classification report

IV. CONCLUSION

The study implemented the Learning Vector Quantization (LVQ) model to classify the ripeness level of palm fruit. Results indicate that the performance heavily depends on the features used. The best performance was achieved by the model using the t-SNE feature with 50% accuracy among all the models.

This illustrates the effectiveness of t-SNE in identifying the ripeness level of palm fruit by extracting relevant features. Conversely, the GLCM feature demonstrated a 40% accuracy on the test data. This indicates that although GLCM has the ability to depict the texture of the palm fruit, it may not wholly embody the characteristics of ripeness. Moreover, the utilization of HSV features resulted in a 45% accuracy. However, this achievement is still inferior to the precision achieved while using t-SNE. The study's findings offer valuable insights into the efficacy of different characteristics when categorizing the ripeness level of palm fruit. The t-SNE feature model stands out as the most suitable option for accomplishing this task with greater accuracy than the other methods.

REFERENCES

[1] D. Anggraeni and A. Hukom, "Analisis Industri Kelapa Sawit Di Kalimantan Selatan Dalam Perspektif Pembangunan Berkelanjutan," *J. Manaj. Ris. Inov.*, vol. 1, no. 2, 2023, doi: https://doi.org/10.55606/mri.v1i2.1074.

[2] D. Levia and Mhubaligh, "Analisis Proses Produksi CPO Untuk Mengidentifikasi Faktor-Faktor Yang Mempengaruhi Kualitas Mutu CPO," *J. Teknol. dan Manaj. Ind. Terap.*, vol. 2, no. 2, pp. 82–89, May 2023, doi: 10.55826/tmit.v2i2.72.

[3] S. Sanjaya, "Penerapan Learning Vector Quantization Pada Pengelompokan Tingkat Kematangan Buah Tomat Berdasarkan Warna Buah," *J. CoreIT J. Has. Penelit. Ilmu Komput. dan Teknol. Inf.*, vol. 5, no. 2, p. 49, Dec. 2019, doi: 10.24014/coreit.v5i2.8199.

[4] A. S. Romadhon and V. T. Widyaningrum, "Klasifikasi Mutu Jeruk Nipis dengan Metode Learning Vector Quantization (LVQ)," *Rekayasa*, vol. 8, no. 2, 2015, doi: https://doi.org/10.21107/rekayasa.v8i2.2065.

[5] W. A. Pulungan, Y. Mulyani, and W. E. Sulistiono, "Identifikasi Kematangan Buah Kopi Menggunakan Jaringan Syaraf Tiruan Learning Vector Quantization," *Barometer*, vol. 4, no. 2, p. 217, Jul. 2019, doi: 10.35261/barometer.v4i2.1834.

[6] M. Effendi, F. Fitriyah, and U. Effendi, "Identifikasi Jenis dan Mutu Teh Menggunakan Pengolahan Citra Digital dengan Metode Jaringan Syaraf Tiruan," *J. Teknotan*, vol. 11, no. 2, p. 67, Oct. 2017, doi: 10.24198/jt.vol11n2.7.

[7] A. Sumarsono and S. Supatman, "Imagery Identification of Tomatoes Which Contain Pesticides Using Learning Vector Quantization," *J. Tek. Inform.*, vol. 2, no. 1, pp. 9–16, Jan. 2021, doi: 10.20884/1.jutif.2021.2.1.15.

[8] M. S. M. Alfatni, A. R. Mohamed Shariff, S. K. Bejo, O. M. Ben Saaed, and A. Mustapha, "Real-time oil palm FFB ripeness grading system based on ANN, KNN and SVM classifiers," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 169, p. 012067, Jul. 2018, doi: 10.1088/1755-1315/169/1/012067.

[9] T. S. Hong, F. Hanim Hashim, T. Raj, and A. B. Huddin, "Classification of Oil Palm Fruit Ripeness Using Artificial Neural Network," in *2021 IEEE International Conference on Automatic Control & Intelligent Systems (I2CACIS)*, IEEE, Jun. 2021, pp. 358–363. doi: 10.1109/I2CACIS52118.2021.9495857.

[10] S. A. Rosiva Srg, M. Zarlis, and W. Wanayumini, "Identifikasi Citra Daun dengan GLCM (Gray Level Co-Occurence) dan K-NN (K-Nearest Neighbor)," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 21, no. 2, pp. 477–488, Mar. 2022, doi: 10.30812/matrik.v21i2.1572.

[11] R. Widodo, A. W. Widodo, and A. Supriyanto, "Pemanfaatan Ciri Gray Level Co-Occurrence Matrix (GLCM) Citra Buah Jeruk Keprok (Citrus reticulata Blanco) untuk Klasifikasi Mutu," *J. Pengemb. Teknol. Inf. Dan Ilmu Komput.*, vol. 2, no. 11, 2018, [Online]. Available: https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/3420

[12] G. Moreira, S. A. Magalhães, T. Pinho, F. N. dos Santos, and M. Cunha, "Benchmark of Deep Learning and a Proposed HSV Colour Space Models for the Detection and Classification of Greenhouse Tomato," *Agronomy*, vol. 12, no. 2, p. 356, Jan. 2022, doi: 10.3390/agronomy12020356.

[13] M. Hall-Beyer, "Practical guidelines for choosing GLCM textures to use in landscape classification tasks over a range of moderate spatial scales," *Int. J. Remote Sens.*, vol. 38, no. 5, pp. 1312–1338, Mar. 2017, doi: 10.1080/01431161.2016.1278314.

[14] J. Ma, "Content-Based Image Retrieval with HSV Color Space and Texture Features," in *2009 International Conference on Web Information Systems and Mining*, IEEE, Nov. 2009, pp. 61–63. doi: 10.1109/WISM.2009.20.

[15] H. Zhou, F. Wang, and P. Tao, "t-Distributed Stochastic Neighbor Embedding Method with the Least Information Loss for Macromolecular Simulations," *J. Chem. Theory Comput.*, vol. 14, no. 11, pp. 5499–5510, Nov. 2018, doi: 10.1021/acs.jctc.8b00652.

[16] R. F. Albuquerque, P. D. L. de Oliveira, and A. P. de S. Braga, “Adaptive Fuzzy Learning Vector Quantization (AFLVQ) for Time Series Classification,” 2018, pp. 385–397. doi: 10.1007/978-3-319-95312-0_33.

[17] A. Sato and K. Yamada, “Generalized Learning Vector Quantization,” *Adv. Neural Netw. Inf. Process. Syst.*, 1996, [Online]. Available: https://papers.nips.cc/paper_files/paper/1995/hash/9c3b1830513cc3b8fc4b76635d32e692-Abstract.html