Detection of Leaf Fall Disease in Sembawa Rubber Plantation Through Feature Extraction Model and Clustering Methods

Alhadi Bustamam[1], Devvi Sarwinda[2], Retno Lestari[3], Ahmad Ihsan Farhani[4], Harum Ananda Setyawan[5], Masita Dwi Mandini Manessa[6], Tri Rappani Febbiyanti[7], Minami Matsui[8]

Department of Mathematics-Faculty of Mathematics and Natural Science, Universitas Indonesia, Depok, Indonesia[1, 2, 4, 5]
Data Science Center UI-Faculty of Mathematics and Natural Science, Universitas Indonesia, Depok, Indonesia[1, 2, 4, 5]
Department of Biology-Faculty of Mathematics and Natural Science, Universitas Indonesia, Depok, Indonesia[3]
Department of Geography-Faculty of Mathematics and Natural Science, Universitas Indonesia, Depok, Indonesia[6]
Indonesian Rubber Research Institute, South Sumatra, Indonesia[7]
RIKEN Tsurumi-ku, Yokohama, Kanagawa, 230-0045, Japan[8]

***Abstract*—Natural rubber is one of Indonesia's most important export commodities, making the country the second-largest exporter globally with a 28.65% share of the world market. However, recent production has declined, partly due to leaf fall disease caused by the *Pestalotiopsis sp.* fungus. This disease leads to premature leaf drop, which forces rubber trees to redirect energy from latex production to leaf regeneration, potentially reducing yields by up to 30%. Traditional detection methods that rely on manual visual inspection of leaf morphology are impractical over large plantation areas. To address this, the present study proposes a remote sensing-based detection approach using aerial drone imagery and unsupervised machine learning. Two feature extraction methods: Convolutional Autoencoder (CAE) and Gray Level Co-occurrence Matrix (GLCM) were used prior to clustering with k-means. Despite a small dataset, the GLCM-based approach significantly outperforms the CAE-based method. These results demonstrate that GLCM combined with clustering can reliably distinguish between healthy and diseased plantation areas. The proposed method offers a cost-effective, scalable, and non-invasive alternative to ground surveys, and has strong potential for real-world deployment in disease monitoring and early warning systems across large agricultural regions.**

Keywords*—Convolutional autoencoder; gray level co-occurrence matrix; k-means clustering; rubber plant plantation; Pestalotiopsis sp

I. INTRODUCTION

The rubber plant (*Hevea brasiliensis*) is a plant in the form of a tall tree and possesses a substantial stem that generates natural rubber. Natural rubber is one among the essential goods for Indonesia to export. From 2014 to 2019, the average export of natural rubber in Indonesia made up to 2.691.120 tons, and Indonesia dominates 28.65% world market share. This makes Indonesia the second biggest natural rubber exporter in the world after Thailand [1].

Recently, natural rubber production in Indonesia has been decreasing. One of the factors causing the decline is the emergence of new leaf fall disease caused by the fungus *Pestalotiopsis sp*. The leaf fall disease causes the rubber tree to drop its leaves prematurely, resulting food reserve of the rubber tree allocated into regrow fallen leaves [2]. It is estimated that the area of rubber plantations in Indonesia was around 3.692.352 hectares in 2021 [3]. The area of rubber plantation that is infected with *Pestalotiopsis sp.* is around 30.328,84 hectares, resulting in a decline of natural rubber production by up to 30% [4].

The disease can be detected through the morphology of the leaf. The visible symptoms on the leaves manifest as brown spots, which subsequently progress into dark brown areas, with a distinct demarcation separating these affected regions from the healthy leaf sections. The size of these spots gradually increases to reach dimensions of 1 to 2 centimeters, eventually leading to necrosis in the surrounding tissue [4].

The ratio of a plant's leaf area to its land surface area is called the leaf area index (LAI) [5]. Research by [6] claims that there is a vigorous negative connection amongst severeness of disease resulting in leaves falling and the leaf zone index. This implies that if the disease is more severe, the leaf area index will also diminish. As the leaf area index decreases, the leaf area is also decreasing, implying that the number of leaves in the tree is also decreasing. When the leaf loss reaches a certain point, the rubber plant will find it difficult to photosynthesis, hence the rubber tree will allocate its food reserves to grow new leaves, rather than using its food reserve to produce latex [6].

When photographed with an aerial drone, the difference between a healthy rubber plantation and an infected rubber plantation is visible. Healthy rubber plantation has dense and tightly closed canopy while infected rubber plantation has sparse and not tightly closed canopy [6]. This gives authors an idea to create a disease detection model using machine learning as an early detection to detect leaf fall disease in rubber plantations particularly caused by the fungus *Pestalotiopsis sp.*, using aerial photos of rubber plantations as the dataset.

Early detection of leaf fall disease offers substantial benefits for rubber farmers and plantation managers.

*Corresponding Author

Identifying affected areas at an early stage enables targeted intervention, such as localized fungicide application or canopy management, which can prevent the disease from spreading further. This minimizes crop loss, preserves yield potential, and reduces unnecessary operational costs associated with broad area treatments. Furthermore, by using aerial imagery and automated analysis, farmers can monitor large plantation areas more efficiently and frequently, improving response times and reducing dependence on labor-intensive ground inspections. Such scalable detection systems can ultimately support more resilient and sustainable rubber production.

Most previous studies on plant disease detection rely on labeled datasets and employ supervised classification methods to identify disease types, typically using close-up images of individual leaves. However, to the best of our knowledge, there is no prior research that utilizes aerial drone imagery to either classify or cluster plantations affected by leaf fall disease. Given that the dataset in this study is unlabeled and collected at the plantation scale using an aerial drone, an unsupervised clustering approach is adopted to detect and differentiate between healthy and infected rubber plantation areas.

Before clustering the infected plantations, the images will be inputted through the feature extraction method. The feature extraction method is used to obtain as much information as possible from the images and to diminish the image's size [7]. The information extracted by feature extraction will then be used as an input for the clustering method to cluster.

The feature extraction used in this research is a convolutional autoencoder (CAE) and gray level co-occurrence matrix (GLCM). Convolutional autoencoder uses a convolution layer and a pooling layer to extract temporal and spatial features from the image [8]. Meanwhile, GLCM uses a matrix, where all elements from the matrix are pairs of pixels that have a brightness level, distance d, and inclination angle θ [9]. These two approaches are applied to cut down the dimension of the picture and extract prominent characteristics from the images.

In this research, the authors use CAE and GLCM as feature extraction methods to cluster healthy rubber plantations and infected rubber plantations, particularly caused by the fungus *Pestalotiopsis sp.* K-means clustering will then be used as a clustering algorithm. Convolutional autoencoder and GLCM are used to reduce the dimension of images and extract prominent features. Then the extracted features will be clustered using k-means clustering, and finally, the result will be evaluated using the silhouette score and davies-bouldin index.

The major contributions of this study are as follows:

- A novel application of unsupervised clustering (k-means) to distinguish between healthy and diseased rubber plantations using aerial drone imagery, without requiring labeled data.
- Implementation and comparison of two feature extraction techniques: CAE and GLCM to identify prominent image characteristics for clustering.
- Empirical validation showing that the GLCM-based approach outperforms CAE in silhouette score and davies-bouldin index, highlighting its suitability for small-sample scenarios.
- A cost-effective, scalable detection framework for early disease identification that supports better decision-making and operational efficiency in large-scale rubber plantations.

This research is organized into five sections: Section I provides the background and motivation for the study. Section II reviews related works and previous research relevant to this topic. Section III describes the dataset, methodology, and models employed in the study. Section IV presents and discusses the results. Finally, Section V concludes the research and outlines potential directions for future work.

II. Related Work

This section outlines previous research related to the current study. A number of past investigations have focused on detecting plant disease using machine learning, and a summary of these studies is presented in TABLE I

TABLE I STATE-OF-THE-ART

Research	Method	Dataset	Evaluation
[10]	MobileNetV3-small	PlantVillage (54,303 images)	Accuracy = 99.5%
[11]	VGG16 (Transfer Learning)	Kaggle Rice Leaf Disease (4,500 images)	Accuracy = 90%
[12]	SVM	PlantVillage (2,152 images)	Accuracy = 98.9%
[13]	SVM + GLCM + k-means	PlantVillage (8,350 images)	Precision = 99%
[14]	SVM + GLCM + LBP + k-means	500+ images (4 disease classes)	Accuracy = 97.2%
[15]	KNN + GLCM + k-means	Carrot leaves (undisclosed, est. ~400)	Accuracy = 95%
[16]	CAE + DenseNet121	PlantVillage (9,920 tomato images)	Accuracy = 98.35%
[17]	Deep KNN (optimized)	Multispectral crop leaf dataset	Accuracy = 89.12%

Khan et al. [10] implemented a MobileNetV3-small model on the PlantVillage dataset consisting of over 54,000 images, achieving an accuracy of 99.5%. Similarly, Jangid [11] utilized transfer learning with VGG16 on a rice leaf disease dataset and reported an accuracy of 90%. Abdu et al. [12] compared support vector machine (SVM) models to CNNs for classifying potato diseases, attaining 98.9% accuracy.

Several studies have integrated traditional feature extraction techniques with machine learning classifiers. Ahmed and Yadav [13] applied SVM in combination with GLCM and k-means clustering on the PlantVillage dataset and achieved 99% precision. Jamjoom et al. [14] enhanced this approach by incorporating local binary patterns (LBP), reaching 97.2% accuracy across four disease types. Similarly, Komala [15] employed a combination of k-means clustering, GLCM, and KNN on carrot leaf images, with an accuracy of 95%.

Recent advancements have also explored the use of deep learning models for feature extraction. Kumar et al. [16] proposed a hybrid approach combining a convolutional autoencoder (CAE) with DenseNet121 for tomato disease classification, reporting an accuracy of 98.35%. In another approach, Gaikwad and Musande [17] utilized a meta-optimized deep KNN model on a multispectral crop dataset, yielding an accuracy of 89.12%.

Previous studies have shown high accuracy in plant disease detection using deep learning and supervised classification approaches. But the methods used are constrained by several limitations. First, most of the research relies on close-up images of individual leaves [10–16], which are impractical for monitoring large plantation areas. Second, these methods generally require large, annotated datasets for training, which are costly and time-consuming to obtain. Finally, many prior works focus on classification into predefined disease types, rather than unsupervised identification of disease-affected areas at the canopy or plantation scale. To address these shortcomings, the present study introduces an unsupervised clustering framework that operates directly on aerial drone imagery without the need for manual labeling. By comparing deep learning-based feature extraction (CAE) with handcrafted texture features (GLCM), we also investigate approaches that can perform robustly under small-sample constraints, making the method more practical for early disease detection in real-world plantation settings.

III. Data and Methodology

This section explains the data and methods used to explore the main topics of this study. Fig. 1 outlines the step-by-step process followed throughout the research.

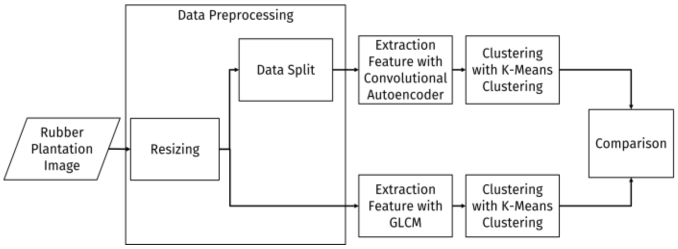

Data Preprocessing
Rubber Plantation Image
Resizing
Data Split
Extraction Feature with Convolutional Autoencoder
Clustering with K-Means Clustering
Extraction Feature with GLCM
Clustering with K-Means Clustering
Comparison

Fig. 1. The research flow of this study.

A. Dataset

The dataset utilized in this study comprises aerial photographs of rubber plantations, collected from the plantation site of PT. Pusat Penelitian Karet Nusantara Sembawa, located in South Sumatra, Indonesia. A total of 19 images were captured using a drone, with each image representing a different section of the plantation and different times, some are taken from 2020 and some are taken from 2021. As illustrated in Fig. 2, the dataset has variation in image perspectives, influenced by differences in drone altitude during image acquisition. Some images were taken from a relatively low altitude, offering a closer view of the canopy (the lower right side of Fig. 2), while others were captured from higher elevations, providing a broader overview of the plantation landscape.

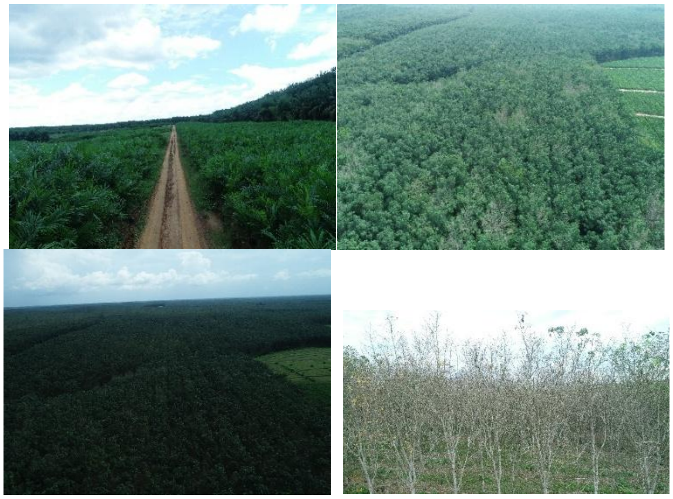


Fig. 2. Sample images of the dataset.

B. Data Preprocessing

- Resizing Data: Resizing images during the data preprocessing stage is a critical step in preparing the dataset for deep learning models. This process involves adjusting the size of images to meet the requirements of the model, with benefits such as computational efficiency, reduced memory load, and ensuring consistency in input sizes for the model, and preventing overfitting. In this study, the image is resized into 20×20 for the input of GLCM and 64×64 for the input of CAE.

- Convert to greyscale: Before applying the Gray Level Co-occurrence Matrix (GLCM) method, the input image must be converted to greyscale. This is because GLCM is designed to analyze texture by evaluating the spatial relationship of pixel intensities, which are represented in a single channel. Color images contain three channels (Red, Green, and Blue), which would complicate the computation. By converting the image to greyscale, it reduces the data to one intensity channel, ensuring that GLCM can accurately process and extract texture features.

- Data Train and Test Split: The division of data into training and test sets is a key stage in the development of models. The training data, which comprises the majority of the dataset, is used to train the model and enable it to understand patterns and features within the data. Meanwhile, test data is employed to assess the model's ability against data it has not previously encountered. In this study, the data was divided into train, validation, and test sets with proportions of 70% and 30%, respectively.

C. Building Models

1) Autoencoder: An autoencoder is a specific type of Artificial Neural Network (ANN) designed to reconstruct its input. Typically, the network's hidden layers engage in dimensionality reduction on the input, learning prominent features that facilitate effective reconstruction [18]. The main idea of the autoencoder is to have the same input size as the output size. Bottleneck layer is then introduced to make the model able to learn the representation of the data. Autoencoder is consist of four components, that is the bottleneck layer, the decoder network, the encoder network and the reconstruction loss [8].

The encoder network is part of the autoencoder that is used to encode input data into a representation of input data. The mathematical equation that represents each layer at the encoder is presented in Eq. (1):

$$X_{e_{i+1}} = f_{e_i}\left(W_{e_i}^T X_{e_i} + b_{e_i}\right) \forall i = 0,1,2,\ldots.,N. \qquad (1)$$

The mathematical notation used in the encoder network is X_{e_i}, $X_{e_{i+1}}$, W_{e_i}, $b_{e_i}, and\ f_{e_i}$, which are the input of i^{th} layer, the output of the i^{th} layer, the weight of the i^{th} layer, bias of the i^{th} layer, and the activation function of the i^{th} layer, respectively. The encoder network's final layer is called the bottleneck layer, and its output is a representation of the input data. As a component of the autoencoder, the decoder layer uses the bottleneck layer's output as input and attempts to remodel input data from its representation. The mathematical equation that represents each layer at the decoder is presented in Eq. (2):

$$X_{di+1} = f_{di}(W_{di}^T X_{di} + b_{di}) \forall i = 0,1,2,\ldots.,N \qquad (2)$$

The mathematical notation in the decoder network is $X_{d_i}, X_{d_{i+1}}, W_{d_i}, b_{d_i}$,and f_{d_i}, which are the input of i^{th} layer, the output of the i^{th} layer, the weight of the i^{th}layer, bias of the i^{th} layer, and the activation function of the i^{th} layer, respectively.

Reconstruction loss is the difference between the original data X^O and the remodelled data X^R. Autoencoder is trained to minimize reconstruction loss. There are two reconstruction losses commonly used, namely binary cross-entropy (BCE) and mean squared error (MSE). The mathematical equation for mean squared error is shown by Eq. (3). Variable D is the number of samples used.

$$MSE(X^O, X^R) = \frac{1}{D}\sum_{j=1}^{D}\left(X_j^O - X_j^R\right)^2 \qquad (3)$$

A convolutional autoencoder operates based on the same principles as a traditional autoencoder, with the key distinction being the utilization of convolutional layers instead of dense layers [19]. Similar to the autoencoder, the CAE also consist of four parts: the encoder, the decoder, the bottleneck layer and the reconstruction loss. While the types layer used by CAE is similar to the convolutional neural network (CNN), it primarily uses a convolutional layer and a pooling layer in the encoder and a convolutional layer and an upsampling layer in the decoder. The type of layer used by CAE makes it more suitable for the image data. As shown in [20], the convolutional layer performance is better when handling the images compared to the fully connected layer due to the number of parameters used in the convolutional layer being less than fully connected layer, making the convolutional layer harder to overfitting and translational invariance. Layers used in the encoder network primarily aims to extract prominent characteristic, while the layers used in the decoder network primarily aims to reconstruct the compressed input image.

Convolutional autoencoders (CAEs) were selected in this study due to their strong capability for unsupervised feature extraction, especially in settings, where labeled data is scarce. Since CAEs are trained to reconstruct input images, they require no manual annotations, the image itself serves as the training signal. By using convolutional layers, CAEs are able to capture local spatial patterns such as edges, textures, and shapes at multiple scales, which are well-suited for analyzing structural variations in aerial plantation images.

The bottleneck layer compresses the input into a low-dimensional latent space, forcing the network to retain only the most essential information. This enables CAEs to act as powerful non-linear dimensionality reduction tools, outperforming traditional linear methods like PCA in capturing complex feature combinations with higher discriminative power [21].

Additionally, by eliminating redundant or noisy pixel information, CAEs help reduce the risk of overfitting, which is particularly advantageous when training on small datasets such as the 19 aerial images used in this research.

2) Gray level co-occurrence matrix: The texture of a gray image is elucidated through the examination of spatial correlation characteristics of gray levels using the gray level co-occurrence matrix [22]. GLCM analyzes textures in images based on grayscale intensity, distance, and angle between representations of two neighboring pixels [23]. Texture features play a crucial role in distinguishing regions within images. They pertain to visual patterns characterized by homogeneity, avoiding the production of a single color or intensity [22]. The gray intensity used in digital image data has a value range of 0 to 255 or a quantization level of 256. The quantization level is the number of possible values that an image will have. Some of the texture metrics that can be used are dissimilarity, contrast, homogeneity, energy, angular second moment (ASM), and correlation.

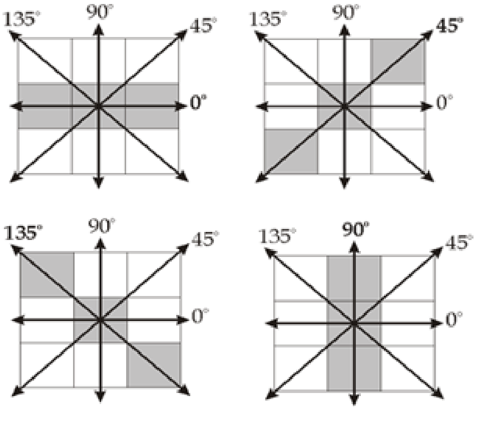


Fig. 3. Angle variation in GLCM.

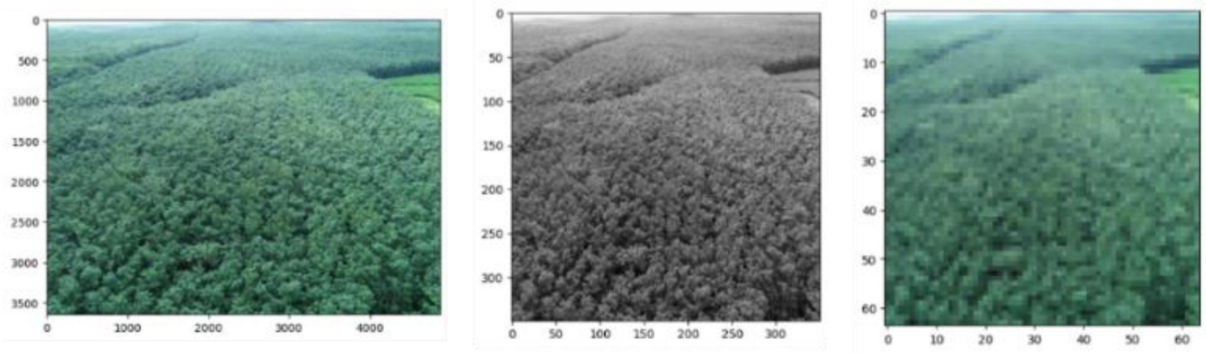
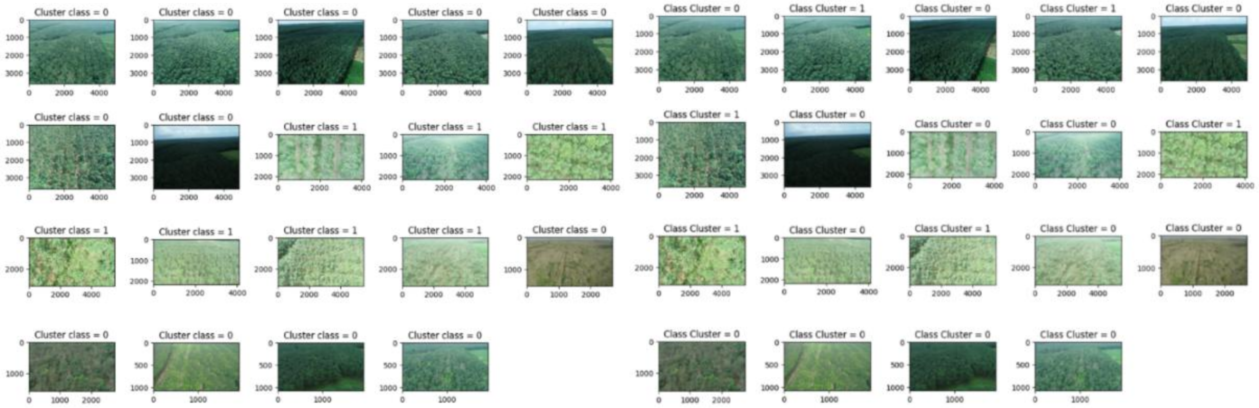


Fig. 4. Comparison between original image (left) and resized image (middle and right).



(a) (b)

Fig. 5. Clustering result for CAE combined with k-means clustering (a), Clustering results of GLCM combined with k-means clustering (b).

Contrast is the magnitude of the difference in color intensity at a pixel coordinate point (x, y) in the image with the surrounding pixels. Dissimilarity is a feature that states the size of the variation in pixel values in the image. Homogeneity is a feature that measures the distribution of each entry value in the GLCM matrix with its diagonal entries. ASM is a measurement of the similarity of a picture. A similar picture will only carry a few gray values; therefore, GLCM will have a slight variation in the value of $(i,)$, but each (i,j) will have a high value. Energy is the square root of ASM, which has a value range from 0 to 1, with 1 representing a constant image or an image whose entry value is the same for each pixel coordinate. Correlation is a measurement of how correlated a pixel coordinate value is with the surrounding pixels. In the GLCM method, there are about eight variations of angles that can be used, namely angles 0°, 45°, 90°, 135°, 180°, 225°, 270°, and 315°, as shown in Fig. 3.

The GLCM is a widely used texture-based feature extraction method that captures spatial relationships between pixels at defined directions and distances. It generates a set of statistical features—such as contrast, correlation, energy, and homogeneity—that quantify texture patterns within an image. These features are particularly effective for differentiating image regions based on their structural properties, making GLCM well-suited for classification and segmentation tasks. Moreover, GLCM is inherently color-independent, allowing it to work efficiently with grayscale images. This makes it a highly appropriate choice for analyzing drone-captured plantation imagery, where subtle variations in canopy texture caused by disease can be more informative than color. In agricultural applications, GLCM has been proven to reliably characterize vegetation patterns and support decision-making in crop monitoring.

3) K-means clustering: K-means clustering is among the clustering approaches in unsupervised learning. This method requires the user to determine the number of clusters, and then the dataset will be divided into clusters. The distance between each centroid point in each cluster and the data points is utilized by k-means clustering to sort out the data. The cluster whose centroid is closest to a given data point comprises the data points. In the k-means clustering method, each data point can only be grouped into one cluster or also known as hard clustering. The following is the algorithm of the k-means clustering according to [24].

(1) Determine the number of clusters (K).

(2) Selects K random points as the centroid (μ_k) per cluster.

(3) Label each data based on the nearest centroid point using the Euclidean distance using Eq. (4), where r_{nk} is the label of each x_n that belongs to cluster K.

$$r_{nk} = \begin{cases} 1, if\ k = \min\limits_{j} \left|\left|x_n - \mu_j\right|\right|^2 \\ \quad 0,\ \text{else} \end{cases} \tag{4}$$

(4) Update the centroid point using Eq. (5) based on the cluster formed by Eq. (4).

$$\mu_k = \frac{\sum_{n=1}^{N}(r_{nk}x_n)}{\sum_{n=1}^{N} r_{nk}} \tag{5}$$

(5) Repeat Steps 3 and 4 until the convergence criteria are fulfilled.

D. *Experimental Design*

This research started by collecting the dataset, which is a rubber plantation image photographed using an aerial drone. The image was then resized into 20×20 and converted into grayscale as GLCM input, while the image resized into $64 \times 64 \times 3$ as CAE input. The dataset will be splitted into train and test dataset for feature extraction using CAE. The image will then be feature extracted with the respective model. Both outputs of the feature extraction model are then clustered using k-means clustering. The clustering result will be evaluated using the silhouette score and davies-boudin index.

The architecture of the convolutional autoencoder (CAE) in this study was designed with a focus on balancing feature richness and overfitting risk, especially given the small dataset (19 images). We used a series of convolutional layers followed by pooling layers in the encoder to progressively reduce spatial dimensions and capture hierarchical patterns in the image. A kernel size of 3×3 was selected to capture fine-grained texture details typical of leaf canopy variation. Pooling layers were used to reduce spatial dimensions while preserving the most salient features. The bottleneck layer was set to 8×8×8 to compress the image while still retaining enough structural information for effective reconstruction and clustering. The decoder mirrored the encoder using upsampling layers to reconstruct the image.

The selection of four principal angles in GLCM 0°, 45°, 90°, and 135°, was made to adequately capture texture orientation diversity across the image. These directions represent the primary spatial alignments typically found in natural and man-made textures, including horizontal, vertical, and diagonal structures. Using these four directions allows the model to extract comprehensive spatial patterns without significantly increasing computational complexity. The features derived from the GLCM—such as contrast, energy, homogeneity, and correlation—statistically represent image texture by quantifying pixel relationships within a local spatial context. These features offer strong descriptive and discriminative power, making them highly effective for a wide range of image analysis tasks, including classification, segmentation, and object detection. In the context of this study, these features help differentiate between dense (healthy) and sparse (infected) canopy textures in aerial plantation imagery.

In this research, the author used two different methods to extract features from photos of rubber plant fields, namely GLCM and a convolutional autoencoder. In the convolutional autoencoder, the data will initially be resized to a size of $64 \times 64 \times 3$. Then the data will be split into training and testing, by testing as much as 30% of the dataset. The CAE architecture used can be seen in Table II. The optimizer used is Adam with a loss function of mean squared error and 200 epochs. In GLCM, first, the data will be resized to a size of 20×20. Then the co-occurrence matrix will be formed. Next, feature extraction will be carried out on the co-occurrence matrix that has been created. After going through the feature extraction, the dataset will be clustered using k-means clustering. For the k-means clustering, the initialization method used is k-means++ to improve and faster convergence. Each experiment was run with 10 different centroid seeds with the maximum number of iterations for each experiment was set to 300. These settings are chosen based on experiment and shows consistent clustering performance and reduced sensitivity to initialization randomness. The results of clustering will be evaluated with a silhouette score and davies-bouldin index.

TABLE II ARCHITECTURE OF CAE

Layer	Layer type	Input shape	Size of kernel	Activation function	Padding	Output shape
1	Input layer	64 × 64 × 3	-	-	-	64 × 64 × 3
2	Convolutional layer	64 × 64 × 3	3 × 3	ReLU	Same	64 × 64 × 16
3	Average Pooling layer	64 × 64 × 16	2 × 2	-	-	32 × 32 × 16
4	Convolutional layer	32 × 32 × 16	3 × 3	ReLU	Same	32 × 32 × 8
5	Average Pooling layer	32 × 32 × 8	2 × 2	-	-	16 × 16 × 8
6	Convolutional layer	16 × 16 × 8	3 × 3	ReLU	Same	16 × 16 × 8
7	Average Pooling layer	16 × 16 × 8	2 × 2	-	-	16 × 16 × 8
8	Convolutional layer (Bottleneck layer)	8 × 8 × 8	3 × 3	ReLU	Same	8 × 8 × 8
9	Upsampling layer	8 × 8 × 8	2 × 2	-	-	16 × 16 × 8
10	Convolutional layer	16 × 16 × 8	3 × 3	ReLU	Same	16 × 16 × 8
11	Upsampling layer	16 × 16 × 8	2 × 2	-	-	32 × 32 × 8
12	Convolutional layer	32 × 32 × 8	3 × 3	ReLU	Same	32 × 32 × 8
13	Upsampling layer	32 × 32 × 8	2 × 2	-	-	64 × 64 × 8
14	Convolutional layer	64 × 64 × 8	3 × 3	ReLU	Same	64 × 64 × 3

IV. Experiments and Result

In this section, we will explain the results of our research and at the same time a comprehensive discussion is also given.

A. The Evaluation Parameters

Silhouette score is an approach applied to assess clustering. The assessment is done by finding the distance between data in one cluster and the distance with other data in different clusters [25]. The mathematical equation for the silhouette score is presented by Eq. (6):

$$S(i) = \frac{p(i)-q(i)}{max(p(i)q(i))} \tag{6}$$

The variable $p(i)$ means the average length of data accompanied by every other data in the similar cluster, while the variable $q(i)$ is the average length of data i with every other data in the nearest cluster, $S(i)$ is the silhouette score of i^{th} data. Silhouette score has a range of values from [-1,1]. A value near 1 denotes an excellent clustering because the data point is closer to its centroid means that each data can be grouped into a cluster that has the same characteristics. Value close to 0 indicates overlapping clusters or the distance between the clusters is not significant. Value close to -1 indicates very poor clustering because the data occupies the wrong cluster [26].

Another metric used to measure the clustering performance is the davies-bouldin Index [27]. Davies -bouldin index measures the average similarity between each cluster and its most similar ones. A lower davies-bouldin index indicates better clustering performance [see Eq. (7)]:

$$DBI = \frac{1}{k}\sum_{i=1}^{k} \max_{j\neq 1}\left(\frac{S_i+S_j}{M_{ij}}\right) \tag{7}$$

where, S_i is the average of intra-cluster distance in cluster i and M_{ij} is the centroid distance between cluster i and j.

B. Results

The outcome of the observations in this study will be discussed here. Fig. 4 displays a comparison of the original and resized images. The original image is shown on the left side, while the resized image is shown on the middle and right sides. The original image in the dataset has a varied shape. The GLCM will reduce the image into 20×20 and convert from an RGB image into a grayscale image, while the CAE will reduce the image into 64×64×3. The output feature extraction for 1 sample in GLCM is a 1D array with a shape of 24, while for CAE, the output size for 1 sample is 8×8×8, but k-means clustering is only able to take input with a shape of a 1D array; the output of CAE will be flattened to transform the output size into 1D array.

The clustering result from CAE combined with k-means clustering is shown in Fig. 5. Clustering result for CAE combined with k-means clustering. It can be seen from 19 samples, 7 of the samples are clustered as infected rubber plant plantation. The clustering results for GLCM combined with k-means clustering is also shown in Fig. 5. From 19 samples, 6 of the samples are clustered as infected rubber plant plantation.

The comparison of silhouette score between CAE and GLCM is also shown. The results of each model reveal that greater the silhouette score value, the better the cluster created by the model. This is because the clusters formed have low distances between data in the same cluster while have large distances between data in different clusters indicating that the clusters formed are well separated. This finding is also supported by the davies-bouldin index of GLCM, which is smaller compared to CAE, indicating better separation. From TABLE IIIresults of each model below, performance of GLCM combined with k-means clustering is better compared to CAE combined with k-means clustering. It means that GLCM combined with k-means clustering has better clustering performance which in this case is separating the healthy rubber plantation and infected rubber plantation.

TABLE III MODELS PERFORMANCE

No.	Method	Silhouette Score	Davies-bouldin Index
1	CAE combined with k-means clustering	0.467	1.638
2	GLCM combined with k-means clustering	0.788	0.653

C. Discussion

The performance difference between the CAE and GLCM methods, reflected in silhouette scores of 0.467 and 0.788, respectively, can be attributed primarily to the limitations of the dataset and the characteristics of each method. Besides, the silhouette scores, clustering quality was further assessed by the davies-bouldin index. The CAE methods achieve 1.638, indicating moderate overlap between clusters and have poor performance on separation, while the GLCM has better performance with the davies-bouldin index of 0.653, indicating well-separated clusters. These findings, based on silhouette score and davies-bouldin index, show that the GLCM provides more discriminative representations resulting in more reliable clustering performance.

One major factor is the small sample size. This study only utilized 19 aerial images, which is far below the typical data requirements for deep neural networks like CAE. As explained by LeCun et al. [28], deep learning models rely on large datasets to learn meaningful representations; otherwise, they risk overfitting and poor generalization. In this case, the CAE likely failed to learn robust feature embeddings due to insufficient training data.

Moreover, the curse of dimensionality further exacerbated the CAE's performance. The output feature map of size 8×8×8 was flattened into a 1D vector of 512 features per image, resulting in a dataset of shape (19, 512). With far more features than samples, the model operated in a highly sparse input space, where distances between points become less meaningful, making clustering less effective [29].

In contrast, GLCM produces a smaller set of statistical texture features that are handcrafted and less sensitive to data volume. These features capture relevant spatial patterns in grayscale images without the need for extensive training. As a result, the GLCM-based clustering method was more robust and better suited to the constraints of a small dataset. This performance gap illustrates the importance of aligning model complexity with data availability: simpler feature extraction techniques may outperform deep learning approaches in low-data scenarios.

This study acknowledges some limitations. The dataset used in this work consists of only 19 images collected from a single plantation, without ground-truth labels or cross-site validation. The clustering results, silhouette score, and davies-bouldin index may be influenced by dataset-specific characteristics rather than genuine disease discrimination. To address this, future research will focus on expanding the dataset to include more than one plantation site and diverse conditions. Ground-truth labels obtained through expert annotation will be incorporated to enable supervised validation.

V. Conclusion

Clustering with images of a rubber plantation as a dataset can be done with a machine learning approach. First, gather images of a rubber plant plantation. Second is to resize the image. Third, extract prominent features from the image using feature extraction. In this study, the feature extraction methods used are CAE and GLCM. Fourth, clustering the extracted image using k-means clustering. The last step is to evaluate the result using the silhouette score and davies-bouldin index. GLCM combined with k-means clustering have better performance compared to CAE combined with k-means clustering, meaning that GLCM combined with k-means clustering is able to separate the healthy rubber plantation and infected rubber plantation.

This study has several limitations. The most prominent being the small dataset size of only 19 aerial images. Such a limited sample restricts the generalizability of the results and poses challenges for deep learning models like the convolutional autoencoder (CAE), which typically require large datasets to learn robust, non-overfitted representations. The risk of overfitting in the CAE is especially high under these conditions, as the model may memorize noise or irrelevant patterns rather than general features of healthy or diseased plantation areas. Additionally, the dimensionality of the CAE's output (512 features) compared to the number of samples introduces sparsity that negatively affects clustering.

Future works should consider data expansion, such as collecting larger and more diverse aerial datasets across different seasons and regions to ensure generalization. Methodologically, hybrid approaches that combine handcrafted features (for example, from GLCM) and deep learning representations (for example, CNN or CAE) should be explored. Alternative approaches of clustering methods, such as Density-Based Spatial Clustering of Application with Noise (DBSCAN) or Gaussian Mixture Model, may better capture complex data patterns. Finally, integrate the proposed framework into a real-time disease monitoring pipeline.

Acknowledgment

This research is funded by Research and Innovation Program Grants for Advanced Indonesia, number: 36/IV/KS/06/2022 and number: 250./PKS/WRII-DISTP/UI/2022 and supported by Data Science Center (DSC) FMIPA UI, Science and Technology Research Partnership for Sustainable Development (SATREPS) Project, Japan International Cooperation Agency (JICA), RIKEN and Indonesia Rubber Research Institute (IRRI).

References

[1] F. Zuhdi, "The Indonesian natural rubber export competitiveness in global market," Int. J. Agric. Syst., vol. 8, no. 2, pp. 130–139, Jan. 2021.

[2] T. R. Febbiyanti, "Mitigasi kehilangan produksi akibat penyakit gugur daun Pestalotiopsis," in Proc. Konferensi Nasional Karet, 2022.

[3] I. Warsito, "Peluang dan tantangan investasi agribisnis karet alam Indonesia," in Proc. Konferensi Nasional Karet, 2022.

[4] N. Damiri, Y. Pratama, T. R. Febbiyanti, S. E. Rahim, D. T. Astuti, and Y. Purwanti, "Pestalotiopsis sp. infection causes leaf fall disease of new arrivals in several clones of rubber plants," Biodiversitas J. Biol. Divers., vol. 23, no. 8, pp. 3943–3949, Aug. 2022.

[5] G. Sitanggang, D. D. Domiri, I. Carolita, and H. Noviar, "Model spasial indeks luas daun (ILD) padi menggunakan data TM-Landsat untuk prediksi produk padi," Semant. Scholar, Oct. 2010.

[6] P. J. K. Alchemi and S. Jamin, "Impact of Pestalotiopsis leaf fall disease on leaf area index and rubber plant production," IOP Conf. Ser.: Earth Environ. Sci., vol. 995, no. 1, Apr. 2022.

[7] W. K. Mutlag, S. K. Ali, Z. M. Aydam, and B. H. Taher, "Feature extraction methods: A review," J. Phys.: Conf. Ser., vol. 1591, Jul. 2020.

[8] P. Bedi and P. Gole, "Plant disease detection using hybrid model based on convolutional autoencoder and convolutional neural network," Artif. Intell. Agric., vol. 5, pp. 90–101, 2021.

[9] M. Yunus, "Feature extraction: Gray level co-occurrence matrix (GLCM)," Medium, Jul. 2020.

[10] A. T. Khan, S. M. Jensen, A. R. Khan, and S. Li, "Plant disease detection model for edge computing devices," Front. Plant Sci., vol. 14, pp. 181–204, Dec. 2023.

[11] S. K. Upadhyay, R. Singh, A. Kumar, and N. Sagar, "Rice Leaves Disease Detection Mechanism Using VGG16 Deep Learning Architecture" Innovations in Data Analytics (ICIDA 2023), vol. 1005, Sep. 2024.

[12] A. M. Abdu, M. M. Mokji, and U. U. Sheikh, "Machine learning for plant disease detection: An investigative comparison between support vector machine and deep learning," IAES Int. J. Artif. Intell., vol. 9, no. 4, pp. 670–683, Dec. 2020.

[13] I. Ahmed and P. K. Yadav, "A systematic analysis of machine learning and deep learning based approaches for identifying and diagnosing plant diseases," Sustain. Oper. Comput., vol. 4, pp. 96–104, Mar. 2023.

[14] M. Jamjoom, A. Elhadad, H. Abulkasim, and S. Abbas, "Plant leaf diseases classification using improved k-means clustering and SVM algorithm for segmentation," Comput., Mater. & Continua, vol. 76, no. 1, pp. 367–382, Jun. 2023.

[15] T. Komala, S. S. Ashwini, and M. Z. Kurian, "Prediction of plant leaf disease using image pre-processing and filter based optimal feature selection for KNN classifier," Int. J. Adv. Res. Ideas Innov. Technol., vol. 7, no. 3, pp. 510-513, 2021.

[16] T. Kumar, et al., "Plant disease detection and classification using hybrid model based on convolutional auto encoder and convolutional neural network," Comput., Mater. & Continua, vol. 83, no. 3, pp. 5219–5234, 2025.

[17] V. P. Gaikwad and V. Musande, "Advanced prediction of crop diseases using Cetalatran-optimized deep KNN in multispectral imaging," Trait. Signal, vol. 40, no. 3, pp. 1093–1106, Jun. 2023.

[18] M. Maggipinto, C. Masiero, A. Beghi, and G. A. Susto, "A convolutional autoencoder approach for feature extraction in virtual metrology," Procedia Manuf., vol. 17, pp. 126–133, 2018.

[19] K. O'Shea and R. R. Nash, "An introduction to convolutional neural networks," arXiv preprint arXiv:1511.08458, Nov. 2015.

[20] P. Pi and D. Lima, "Gray level co-occurrence matrix and extreme learning machine for Covid-19 diagnosis," Int. J. Cogn. Comput. Eng., vol. 2, pp. 93–103, Jun. 2021.

[21] H. Niu, G. B. McCallum, A. B. Chang, K. Khan, and S. Azam, “Exploring unsupervised feature extraction algorithms: Tackling high dimensionality in small datasets,” Sci. Rep., vol. 15, no. 1, pp. 21973, 2025.

[22] H. Halim, J. Hendryli, and D. E. Herwindiati, “Online product search using gray level co-occurrence matrix, color moments, and histogram of oriented gradients for content based image retrieval,” IOP Conf. Ser.: Mater. Sci. Eng., vol. 852, Jul. 2020.

[23] K. Trang, L. TonThat, and G. Minh, “Plant leaf disease identification by deep convolutional autoencoder as a feature extraction approach,” in Proc. Int. Conf. Electrical Eng./Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), pp. 522-526, Jun. 2020.

[24] C. Bishop, Pattern Recognition and Machine Learning. New York, NY, USA: Springer, 2006.

[25] P. J. Rousseeuw, “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis,” J. Comput. Appl. Math., vol. 20, pp. 53–65, Nov. 1987.

[26] D. L. Davies and D. W. Bouldin, “A cluster separation measure,” IEEE Trans. Pattern Anal. Mach. Intell., vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979.

[27] Y. Januzaj, E. Beqiri, and A. Luma, “Determining the optimal number of clusters using silhouette score as a data mining technique,” Int. J. Online Biomed. Eng., vol. 19, no. 4, pp. 174–182, Apr. 2023.

[28] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, May 2015.

[29] N. Venkat, “The curse of dimensionality: Inside out,” RG Open, 2020.