Journal of Social Research

P-ISSN:2827-9832
E-ISSN: 2828-335x
Vol. 5, No. 6, May 2026
http://ijsr.internationaljournallabs.com/index.php/ijsr

Analysis of the Combination of K-Nearest Neighbor (KNN) and K-Means for the Classification of Rice Leaf Diseases Using Image Segmentation Method

Rizal Afandi*, Kartini, Retno Mumpuni
Universitas Pembangunan Nasional, Indonesia
Email: 19081010146@student.upnjatim.ac.id*, kartini.if@upnjatim.ac.id, retnomumpuni.if@upnjatim.ac.id

Keywords
rice leaf diseases, image segmentation, k-means clustering, k-nearest neighbor, machine learning

Abstract
Rice is a staple food crop in many Asian countries, including Indonesia, and plays a crucial role in ensuring food security and economic stability. However, rice production is significantly threatened by various leaf diseases, which can lead to substantial yield losses if not detected early. Traditional disease identification methods rely heavily on manual observation, making them inefficient and prone to errors, particularly in early stages when symptoms are less visible. This study aims to develop an automatic classification system for rice leaf diseases by combining K-Means clustering and K-Nearest Neighbor (KNN) algorithms using image segmentation techniques. The research utilized a dataset of 5,932 rice leaf images obtained from Kaggle, consisting of four disease categories: Bacterial Blight, Blast, Brown Spot, and Tungro. The methodology involved preprocessing steps including grayscale conversion, image resizing, and feature extraction, followed by segmentation using K-Means and classification using KNN with various K values. The results indicate that the combination method provides effective classification performance, with the highest accuracy achieved at 88% under a 90:10 training-testing data ratio. Additionally, smaller K values tend to yield better accuracy compared to larger ones. Despite some misclassification between visually similar diseases, the proposed model demonstrates strong potential for practical application. This system can assist farmers in early disease detection, reduce crop losses, and contribute to improving agricultural productivity through data-driven approaches.

INTRODUCTION

Rice plants are the main source of food for most people in Asia, especially in Indonesia, where rice is a staple commodity that supports the nutritional needs of the community (Harlina et al., 2023; Mohidem et al., 2022; Rozi et al., 2023; Sitaresmi et al., 2023). As a crop that is widely cultivated in lowland areas, rice has strategic value in food security and economic stability of agricultural communities (Akanbi et al., 2024; Jin & Zhong, 2022; Sutardi et al., 2022). Rice crops also show resistance to various extreme conditions such as weather changes, pest attacks and plant diseases, which are significant challenges in rice cultivation. Although rice varieties have evolved through a process of adaptation and selection over thousands of years, the productivity of these crops is still often threatened by various types of diseases. According to a report from *the International Rice Research Institute* (IRRI), pest and disease attacks result in crop losses of up to 37% per year in the Asian region, which of course has a major impact on food availability in the region. To mitigate these impacts, the development of high-resolution satellite technology and vegetation indexes to monitor the condition of rice

crops provides a great opportunity in improving productivity and crop cultivation management (Reengineering et al., 2022a)(Vikram & Sarkar, 2022).

However, the challenge of identifying diseases in rice plants in the field is still a significant obstacle, especially for farmers who do not have access to or in-depth knowledge of the symptoms of plant diseases (Jafar et al., 2024; Khan et al., 2025; Nizamani et al., 2023; Tasfe et al., 2024). The manual rice disease identification process relies on expertise and visual observation of disease symptoms, such as changes in leaf color and texture. This leads to less efficient identification and is often prone to error, especially in the early stages of disease development where the symptoms are not clearly visible. Inaccuracies in identification can exacerbate the spread of the disease, potentially causing huge losses to farmers. Therefore, the development of technology that is able to support the automatic diagnosis of plant diseases is needed to help farmers in preventing and handling diseases more effectively (Bhargava et al., 2024; Li & Wang, 2024; Negi & Anand, 2024).

The rapid development of information and communication technology (ICT) offers a promising solution to this problem, especially through the introduction of digital image-based objects and big data analysis. In the context of agriculture, big data processing technologies such as data mining have proven to be effective in the process of extracting important information from large amounts of data. Through clustering and classification techniques, data mining allows the grouping and classification of plant diseases based on the similarity of visual characteristics, such as color, texture, and patterns in infected rice leaves. This allows the detection system to recognize disease patterns automatically, making it easier to analyze and classify. This technology also improves the accuracy of plant disease identification by recognizing specific visual patterns that are difficult for humans to identify. This approach offers speed and efficiency in disease detection, which is expected to improve a more structured plant disease management system.(Worung et al., 2020)

In addition, the use of clustering and classification algorithms, such as K-Means and K-Nearest Neighbor (K-NN), can enrich the disease classification system in rice plants with a higher level of accuracy. K-Means is a non-hierarchical grouping method that serves to divide data into several groups or clusters based on common characteristics. In the context of plant disease management, K-Means can be used to group various disease symptoms based on visual patterns on leaves, thus facilitating the disease identification process. This algorithm works by dividing the data into a number of clusters, where each cluster represents a specific characteristic pattern. K-Means is particularly useful in this process due to its ability to handle data with varied visual characteristics, making it easier to distinguish different types of plant diseases.(Widyanti et al., 2023)

On the other hand, the K-Nearest Neighbor (K-NN) algorithm is used as a classification method that relies on historical data or pre-classified data. This algorithm works based on the proximity between the new data and the existing training data, which allows the determination of disease categories in plants based on the visual characteristics of similar leaves. Previous research has shown that K-NN is effective in classifying image-based objects with high accuracy, especially on large data that requires precision in pattern recognition. K-NN has an advantage in its ability to handle data that is susceptible to noise and visual variation, which makes it particularly relevant for image-based plant disease classification applications. In the rice disease detection system, K-NN can function to identify the types of diseases in rice leaves

that have been grouped by the previous K-Means algorithm, so that the combination of these two algorithms is expected to provide more accurate and effective classification results (Azzahra Nasution et al., 2019).

The increasing need for technology-based solutions makes the combination of K-Means and K-NN have the potential to increase food security in Indonesia. This digital image-based early detection technology not only helps farmers in recognizing disease types quickly and accurately, but also supports governments and stakeholders in the agricultural sector in formulating better policies for the management of agricultural resources. On the other hand, the application of this technology opens up opportunities for further research in the field of artificial intelligence (AI) and machine learning for the agricultural sector, which will expand insights into the management of plant diseases in a more integrated and measurable manner.

In addition, this research is expected to make a major contribution to the development of image processing technology for the agricultural sector, especially in developing countries such as Indonesia. By applying the K-Means and K-NN methods that have been proven to be effective in the classification of image-based objects, this research is expected to create a plant disease detection system that is not only accurate but also easy to implement in the field. This technology is expected to help farmers in managing rice crop diseases, reduce the risk of losses due to diseases, and support efforts to increase national food security through applicable and sustainable technology.

Previous studies have strengthened the reliability of the K-Means and K-NN methods in image-based object classification, particularly in agricultural applications. For example, the study showed the effectiveness of the K-NN method in identifying the level of ripeness of Carica Papaya fruit based on RGB values with high accuracy. The results of the study provide the basis that K-NN is very relevant for visual-based classification that relies on color patterns in images. In addition, the research utilizes K-Means to group rice types based on RGB color features with an accuracy rate of 77.5%. These findings demonstrate the effectiveness of the clustering method in grouping objects based on relevant visual similarities in the classification of plant diseases(Suban et al., 2020)(Trisnawan et al., 2019).

The research also proves the effectiveness of K-Means in grouping data with complex color variations, such as the classification of mango ripeness. The use of color transformation and feature extraction allows the system to recognize the visual characteristics of the fruit with a high degree of accuracy, demonstrating the ability of this algorithm to handle objects with diverse color patterns. Other studies have shown that the combination of K-Means and K-NN can be used effectively to identify citrus leaf disease, with a disease classification accuracy of up to 90.83%. These results prove that the combination of grouping and classification can result in a more precise classification system in recognizing plant diseases.(Premana & Pandhu Wijaya, 2022).

The development of machine learning in agriculture has opened new opportunities for automatic plant disease detection. Image-based classification systems allow disease symptoms to be identified through visual features such as color, texture, intensity, and lesion pattern. Previous research has shown that rice leaf disease classification can be performed using several algorithms, including Support Vector Machine, Random Forest, Artificial Neural Network, CNN, KNN, and clustering-based approaches. For example, recent rice disease classification

research using machine learning and segmented images has demonstrated that computational models can distinguish healthy and diseased leaves with promising accuracy.

Several previous studies provide a relevant foundation for this research. Research on rice leaf image segmentation using K-Means clustering and GLCM features reported classification accuracy values of around 85.71% for bacterial leaf blight, 86% for brown spot, and 83.6% for leaf smut. Another study using KNN based on HSV color and GLCM texture features also confirmed that KNN can be applied to classify rice plant diseases. In addition, comparative research on rice leaf disease classification has shown that deep learning models may reach very high accuracy, but they often require larger computational resources and more complex model architectures.

Although deep learning approaches have shown strong performance, simpler machine learning methods remain relevant because they are easier to implement, interpret, and adapt in resource-limited agricultural environments. K-Means is useful for grouping image regions based on similar visual characteristics, while KNN classifies new data by measuring similarity to labeled training data. The combination of these two methods is therefore logically suitable for rice leaf disease classification because segmentation can simplify the image structure before classification is performed. The manuscript also emphasizes this logic by using K-Means to group image features and KNN to classify disease categories based on the transformed image data.

The research gap lies in the need for a practical classification model that combines segmentation and classification while remaining computationally simple. Many studies focus either on advanced deep learning architectures or on single-algorithm classification, but fewer studies emphasize the combined use of K-Means and KNN for rice leaf disease classification with multiple disease classes and different training-testing scenarios. Moreover, diseases such as blast and brown spot can have visually similar symptoms, which increases the risk of misclassification. This gap indicates that further testing is needed to evaluate whether a segmentation-classification combination can improve recognition patterns and produce stable accuracy across different data split ratios.

The urgency of this research is strengthened by the need to support farmers with early, accessible, and data-driven disease detection tools. Inaccurate disease identification can delay treatment decisions, increase disease spread, and reduce crop yield. A digital image-based classification system can help farmers and agricultural extension workers identify disease types more quickly, especially when expert diagnosis is unavailable. In Indonesia, where rice is a major staple food and many farming communities still rely on manual observation, this kind of technology can contribute to more responsive disease management and strengthen food security.

The novelty of this research is found in the integration of K-Means clustering and KNN classification using image segmentation for four rice leaf disease categories. The study does not merely classify images directly, but first applies preprocessing and segmentation to structure the visual data before classification. It also evaluates several K values and training-testing ratios, allowing the model performance to be observed under different experimental conditions. This provides a more systematic understanding of how segmentation quality, training data proportion, and neighborhood parameters influence rice disease classification accuracy.

Therefore, the purpose of this research is to develop and evaluate an automatic rice leaf disease classification model using the combination of K-Means and KNN with image segmentation. The main objective is to determine how effectively the model classifies Bacterial Blight, Blast, Brown Spot, and Tungro based on digital leaf images. The contribution of this research lies in offering a simpler and applicable machine learning approach for agricultural disease detection, while its practical benefit is to assist farmers, researchers, and agricultural stakeholders in reducing crop losses through faster and more accurate disease identification.

RESEARCH METHODS

To obtain information and data needed in data collection, use the following techniques:

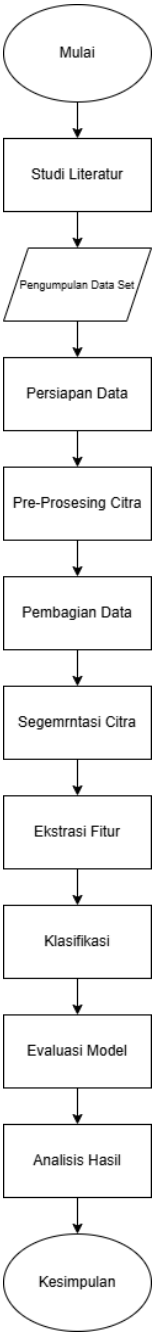

Figure 1. Research Methodology Diagram

Literature Studies

At this stage, a search and collection of references and literature is carried out. The study stage is carried out to obtain information used as a reference, the literature study stage involves reading scientific journals, reference books, previous research, and other sources related to the research topic. This method uses literature research from various fields of science related to program suggestions and workflows:

1. Types of diseases in rice plants
2. Digital image processing
3. Concept of the K-Nearest Neighbor (K-NN) algorithm for Classification of diseases in rice leaves
4. Concept of the K-Nearest Neighbor (K-NN) algorithm for Classification of diseases in rice leaves
5. Pyhton Programming

In addition to being able to get explanations related to the concepts and definitions of a foreign method and terms, from this stage the author can also find gaps from previous research and get several options on how to design a system. Based on the description above, it can be concluded that this stage of literature study is a fairly important stage in the research process.

Data Set Collection

In this study, data collection was carried out using a data set taken from *the Kaggle* website with the name of the user owner NIRMAL SANKALANA which contains various diseases in rice leaves. The data is public and consists of 4 types of data, namely *Bacterial Blight* disease with 1584 images, *Blast* disease with 1440 images, *Brown spot* disease with 1600 images, and *Tungro* disease with 1308 images in JPG format with a total dataset of 5932 datasets. A total of 5932 leaf image data were collected and divided into 2000 test data and 3932 training data, consisting of 500 types of leaf diseases each with the categories *of Bacterial Blight*, *Blast*, *Brown Spot*, and *Tungro.* Then, 3932 training data was also divided into 983 for each category (*Bacterial Blight*, *Blast*, *Brown Spot*, and *Tungro).* After that, *cropping* according to the *background* of the aids. The following is an example of a sample of data that will be used:

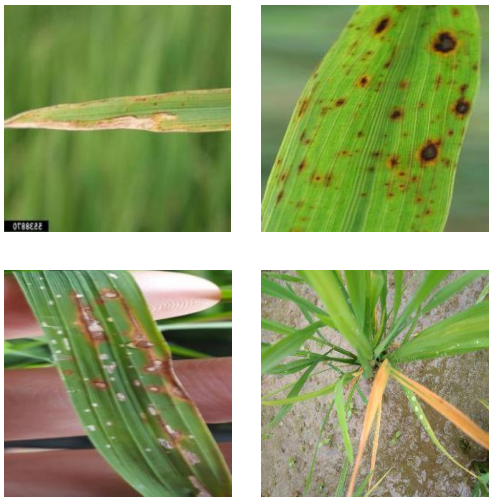

Figure 2. Sample Data used

System Planning

The design is needed in this study to segment the image using the *K-Means* algorithm and the rice leaf disease classification process. The first stage is to *input* data on rice leaf disease images. The second stage *pre-processed* image data of rice leaf disease using digital image processing methods. The third stage is to segment rice leaf disease using the *K-Means* algorithm. And the fourth stage is classification using the K-NN algorithm.

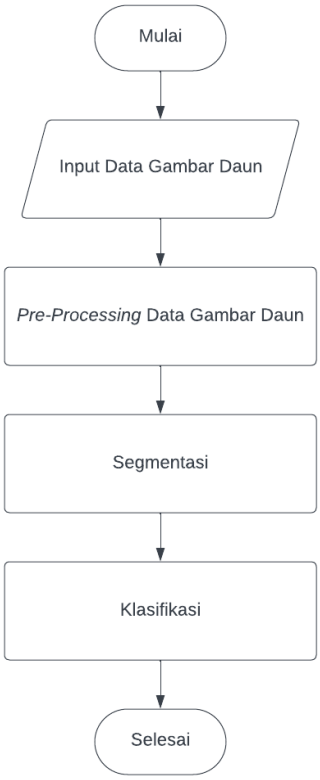

Figure 3. *Program Planning* Flowchart

Pre-processing of images

In the image preprocessing stage, the previously collected datasets will be processed to prepare the data before segmentation and classification. This stage aims to simplify image data, reduce complexity, and ensure that the data used has a uniform format so that it can be processed properly by algorithms.

In this study, image preprocessing is carried out through several stages, namely image conversion to grayscale, image resizing, and image transformation into one-dimensional vector. These stages are done sequentially to ensure that each image has the same characteristics before moving on to the next stage.

Grayscale

At this stage, the original warrant format (RGB) image of the rice leaf is converted into a grayscale image. This process aims to simplify image information by converting three warrant channels into one intensity channel, thereby reducing data complexity and speeding up the computational process.

In this study, the grayscale conversion process was carried out using the OpenCV library with the cv2.cvtColor function, where the image was converted from RGB format to grayscale. Each image that is successfully loaded will go through this process before proceeding to the next stage.

Using grayscale imagery, the system can focus more on the intensity patterns that characterize rice leaf disease, such as differences in brightness levels in infected areas compared to healthy areas.

Resizing

After the grayscle conversion process is done, the next stage is image resizing. Resizing is done to uniformly size all images in the dataset so that they have the same dimensions, thus simplifying the data processing process.

In this case, each image is resized to 64x64 pixels using the cv2.resize function. The selection of this size aims to maintain a balance between image detail and computational efficiency. Too large can slow down the processing process, while too small can remove important information from the image.

With a uniform size, the algorithm used can work more optimally because each data has the same number of features.

Data Transformation (Flattening)

After the preprocessing stage is completed, the next step is image segmentation using the K-Means algorithm. In this study, K-Means was used to group image data based on the similarity of characteristics.

The segmentation process was carried out by determining the number of clusters as many as 4, which was adjusted to the number of rice leaf disease classes used in this research, namely Bacterial Blight, Blast, Brown Spot, and Tungro. The K-Means algorithm will group the data based on the closest distance to a predetermined centroid.

In its implementation, image data that has gone through the preprocessing stage will be used as input for the clustering process. The result of this process is in the form of grouping data which will then be used as input at the classification stage using K-Nearest Neighbor (KNN).

With this segmentation process, it is hoped that data that has similar characteristics can be grouped properly so that it can increase accuracy at the classification stage.

Image Segmentation Using K-Means

After the preprocessing stage is completed, the next step is image segmentation using the K-Means algorithm. In this study, K-Means was used to group image data based on the similarity of characteristics.

The segmentation process was carried out by determining the number of clusters as many as 4, which were adjusted to the number of rice leaf disease classes used in this study, namely Bacterial Blight, Blast, Brown Spot, and Tungro. The K-Means algorithm will group the data based on the closest distance to a predetermined centroid.

In its implementation, image data that has gone through the preprocessing stage will be used as input for the clustering process. The result of this process is in the form of data grouping which will then be used as input at the classification stage using K-Nearest Neighbor (KNN).

With this segmentation process, it is hoped that data that has similar characteristics can be grouped properly so that it can increase accuracy at the classification stage.

Classification Using K-Nearest Neighbor (KNN)

The next stage after the segmentation process is classification using the K-Nearest Neighbor (KNN) algorithm. This algorithm is used to determine the class from the image data based on the proximity to the trained data that the label has known.

In this study, the data from the transformation from K-Means will be used as input to the KNN algorithm. The classification process is carried out by comparing the distance between the test data and the training data using a certain K value.

The K value in this study was tested with several variations, namely from K = 1 to K = 10, to determine the optimal K value based on the best accuracy results. The smaller the K-value, the more sensitive the model will be to the training data, while the larger K-value results in more general decisions.

The results of this classification process are in the form of predictions of the disease class of rice leaves which will then be evaluated at the testing stage.

Implementation

The implementation stage is the stage of system development based on previous designs. In this study, the implementation was carried out using the Python programming language by utilizing several libraries such as OpenCV, NumPy, and Scikit-learn.

All stages from image preprocessing, segmentation using K-Means, to classification using KNN are integrated into one system that is able to identify rice leaf diseases automatically.

Testing and Analysis

The first stage of testing was carried out by paying attention to the *value of the Scale Factor* in the *reescaling* process. To determine the *best Scale Factor* value , it is done by looking for the best accuracy results in several *Scale Factor value experiments*.

Second, using the *Silhouette Coefficient method* to determine the best value of the number of *clusters* in the segmentation using *K-Means*.

Third, conducting tests to obtain optimal K-values in processing using the K-NN algorithm. To get the optimal K value, testing is carried out using the method of finding the best accuracy results in experiments with several K values.

The last one analyzed the program accuracy value from the disease identification results using the K-NN algorithm which was generated through the image segmentation process using the *K-Means algorithm*.

RESULTS AND DISCUSSION

Program Implementation

This subchapter explains the implementation of programs that use K-Means and *K-Nearest Neighbor (KNN)* to identify images of rice leaf disease. This will be explained in the sub-chapters according to the process carried out below.

Dataset Preparation

The data used in this research uses data from Kaggle. The data is public data consisting of four classes, namely *Bacterial blight, blast, brownspot,* and *tungro*. The dataset has a total of 1584 images, 1444 images of Blast disease, 1440 images of Brown Spot, 1600 images of Brown Spot, and 1308 images of Tungro disease in JPG format with a total dataset of 5932 datasets. The image data is then collected into a folder that has been differentiated by class and

then uploaded to Google Drive to facilitate the processes that will be carried out next as seen in figure 4.

Figure 4. Folder Dataset

Data Preprocessing

In the pre-processing stage of data, the datasets that have been collected previously are pre-processed data. This process is carried out by uploading data, changing the image data form and extracting features into Grayscale format, after which the image dataset will be divided according to class.

```
# Mount Gdrive and Load Dataset
from google.colab import drive
drive.mount('/content/drive')

folder_path = "/content/drive/MyDrive/Thesis/Sample"
```

Program Code 4.1. Program code loads data set

Program Code 4.1, specifies the directory of the stored image dataset in this case is set into the storage that has been prepared on google drive "/content/drive/MyDrive/Thesis/Sample". In the source code above, there is a code to connect between google colab and google drive so that the data called will be loaded or processed further.

```
# Functions for image processing (grayscale, resizing, flattening)

def load_and_process_images(folder, num_images, img_size=(64, 64)):

    images = []

    filenames = os.listdir(folder)[:num_images]

    for     filename     in     tqdm(filenames,     desc=f"Processing
{folder.split('/')[-1]}"):

        img = cv2.imread(os.path.join(folder, filename))

        if img is not None:

            gray_img = cv2.cvtColor(img, cv2. COLOR_BGR2GRAY)

            resized_img = cv2.resize(gray_img, img_size)

            flattened_img = resized_img.flatten()

            images.append(flattened_img)

    return np.array(images)
```

Program Code 4.2. Data preprocessing program code

Program code 4.2 is an important part of the steps to load the image dataset in .jpg format. In the above program snippet, the 'load_and_process_images' function is used to load and process images in *grayscale* from the specified directory. This function not only loads the image, but also converts the image to grayscale, resizes it, as well as flattens the image for use in further analysis. This function accepts three parameters: the folder as the directory path where the images are stored, num_images to specify the number of images to be loaded, and img_size to specify the size of the image after it has been resized (the default is 64x64 pixels).

The process starts with initializing a blank list of images that will store the processed images. The function then retrieves a number of file names according to the num_images of the directory specified by the folder. During the iteration, the images in each file are read using cv2.imread. If the image loads successfully, it is converted to grayscale using cv2.cvtColor with the cv2 flag. COLOR_BGR2GRAY. This grayscale image is then resized according to img_size parameters using cv2.resize. Next, the resized image is flattened into a one-dimensional vector using the flatten method. These flattened images are then added to the list images.

To provide feedback to the user during the process, the iteration is wrapped with the tqdm function, which displays a progress bar. Once all the images have finished processing, the list images are converted into a NumPy array and returned as the output of the function.

Once all selected images are loaded, the list images are returned as the output of the function. This image conversion to grayscale format is done to simplify data and reduce memory requirements.

```
# Loading and Processing data

class1_images    =    load_and_process_images(os.path.join(folder_path,
"Bacterial blight"), 1584)
class2_images    =    load_and_process_images(os.path.join(folder_path,
"Blast"), 1440)
class3_images    =    load_and_process_images(os.path.join(folder_path,
"Brownspot"), 1600)
class4_images    =    load_and_process_images(os.path.join(folder_path,
"Tungro"), 1308)
```

Program Code 4.3. Program code loads datasets and processes data

Program code 4.3 is an important part of the steps to load datasets in grayscale format. The process of loading images in *grayscale* for the four classes of rice leaf diseases, namely *Bacterial Blight, Blast, Brown Spots,* and *Tungro* loaded and processed using load_and_process_images functions. This function not only loads images from the appropriate directory for each class, but also performs a series of image processing operations, such as conversion to grayscale, image resizing to 64x64 pixels, and leveling the image into a one-dimensional vector.

The drawings from the Bacterial Blight, Blast, Brown Spot, and Tungro classes amount to 1584, 1440, 1600, and 1308 drawings, respectively. Each image class is processed separately and stored in class1_images, class2_images, class3_images, and class4_images variables. The load_and_process_images function ensures that each image loaded from the associated directory is grayscaled, resized, and flattened into a one-dimensional vector before being saved.

```
# Combining data and labels

all_images = np.vstack((class1_images, class2_images, class3_images,
class4_images))

all_labels = np.array([0]*1584 + [1]*1440 + [2]*1600 + [3]*1308)
```

Program Code 4.4. Combine data and labels

Program code 4.4 is an important part of the loading and image processing process of the four classes of rice leaf diseases, namely Bacterial Blight, Blast, Brown Spot, and Tungro. This process is done through the load_and_process_images function, which loads an image of each class, converts it to grayscale, resizes it to 64x64 pixels, and flattens it out into a one-dimensional vector. With these steps, each image of each class is processed in such a way that it is ready for further analysis.

Once the images of each class have been successfully processed, the 'all_labels' function plays a role in combining all the images into a single unit and labeling each class of images. Images from all four classes are combined into one large array using np.vstack. Numerical labels are then assigned to each image according to their class using np.array, with labels 0 for Bacterial Blight, 1 for Blast, 2 for Brown Spot, and 3 for Tungro.

Data Process

In the data processing stage, the dataset that has previously gone through data pre-processing will be processed next. This process begins with testing data, applying the k-means clustering model, applying the k-nn clustering model, and visualizing the prediction results.

```
# Functions to run various test scenarios

def run_all_scenarios(all_images, all_labels):

    results = []

    split_ratios = [0.1, 0.2, 0.3, 0.4]


    For test_size in split_ratios:

        train_images,   test_images,   train_labels,   test_labels   =
train_test_split(

            all_images,         all_labels,         test_size=test_size,
random_state=42)


        silhouette_score_train = []

        silhouette_score_test = []
```

Program Code 4.5 Running the various scenarios to be tested

Program code 4.5 is an important part of running various test scenarios on processed image data. The *run_all_scenarios* function tests the model using various ratios of the division of the training data and the test data specified in split_ratios variables (0.1, 0.2, 0.3, and 0.4). For each

division ratio, the data is divided into training data and test data using *train_test_split*. The *train_images* and *test_images* variables store the images used for *Training* and testing, while *train_labels* and *test_labels* keep the appropriate labels.

Next, this function initializes two lists, *silhouette_score_train* and *silhouette_score_test*, to store the Silhouette Score values of the practice data and test data. For each K value from 1 to 10, the K-Means model is used to group the train data into 4 clusters. The Silhouette Score is calculated for training data and test data based on the clustering results. The K-Nearest Neighbors (KNN) model was then trained using the results of the K-Means transformation from the training data and tested on the test data. The accuracy of the prediction is calculated and all test results are stored in the results list. In addition, the visualize_predictions function is called to display a picture of the predicted results, providing a clear visualization of the model's performance.

```
# K-Means Clustering

kmeans = KMeans(n_clusters=4, random_state=42)

kmeans.fit(train_images)

# Silhouette Score

silhouette_score_train.append(silhouette_score(train_images,
kmeans.labels_))

silhouette_score_test.append(silhouette_score(test_images,
kmeans.predict(test_images)))
```

Program Code 4.6 Implementation of K-Means clustering

Program code 4.6 describes the application of the K-Means Clustering method to training data. In this section, the K-Means model is initialized with the n_clusters=4 parameter to match the number of clusters to the number of rice leaf disease classes. This model is then trained using training data (*train_images*). After clustering is complete, the Silhouette Score is calculated to measure the quality of the cluster on the training data and test data. This Silhouette Score provides information on how well the training and test data are clustered, with the results stored in *silhouette_score_train* and *silhouette_score_test* lists. This value is useful for assessing the quality of clustering carried out by the K-Means model.

```
# K-Nearest Neighbors (KNN)
knn = KNeighborsClassifier(n_neighbors=k)
knn.fit(kmeans.transform(train_images), train_labels)
predictions = knn.predict(kmeans.transform(test_images))

accuracy = accuracy_score(test_labels, predictions)
results.append({
     "split_ratio": f"{(1-test_size)*100:.0f}:{test_size*100:.0f}",
     "K": k,
     "accuracy": accuracy,
     "silhouette_score_train": silhouette_score_train[-1],
     "silhouette_score_test": silhouette_score_test[-1]
              })
```

Program Code 4.7 Application *of K-Nearest Neighbor (KNN)*

Program code 4.7 functions to apply the K-Nearest Neighbors (KNN) method to data after the K-Means clustering process. After the K-Means model is trained, the training data that has been transformed by K-Means is used to train the model *K-Nearest Neighbor (KNN)* with a range of K values (from 1 to 10). Models *K-Nearest Neighbor (KNN)* trained with trained data that has been clustered and tested on test data that has also been transformed with K-Means. Accuracy of model predictions *K-Nearest Neighbor (KNN)* calculated using *accuracy_score*, and the results are stored in the results list. This accuracy assessment gives an idea of how well the model is *K-Nearest Neighbor (KNN)* in classifying test data after the data has been grouped by K-Means.

```
# Visualization of predictions
visualize_predictions(test_images, test_labels, predictions, test_size,
k)
    return to pd. DataFrame(results)
```

Program Code 4.8 Prediction Visualization

Program code 4.8 involves visualizing the predicted results to provide a visual picture of the model's performance. The *visualize_predictions* function is used to display an image of the predicted result along with the original label on each test scenario. This visualization is performed for each combination of the data sharing ratio and the K-value, facilitating the evaluation and interpretation of the performance of the K-Means model and *K-Nearest Neighbor (KNN)*. The images displayed provide direct insight into the predicted quality and suitability of the predicted label with the original label. The results of this visualization, along with the data collected from the test results, are stored in the form of DataFrame pandas for further analysis.

Test Results

The test was carried out to evaluate the performance of the combination model of the K-Means and *K-Nearest Neighbor (KNN) algorithms* in the classification of rice leaf diseases. The evaluation process involves several ratios of the division of training data and test data, namely 90:10, 80:20, 70:30, and 60:40, as well as testing with various K parameter values on the *K-Nearest Neighbor (KNN)* algorithm, ranging from K = 1 to K = 10, to determine the optimal K value. This test aims to assess the algorithm's ability to accurately classify data and analyze misclassification errors that occur through matrix *confusion*.

The test results consist of two main components, namely *confusion matrix* and prediction visualization. *Confusion matrix* is used to provide a quantitative picture of model performance, while prediction visualization presents a visual sample of data, complete with original labels and model prediction results. This evaluation is expected to provide insight into the combination of parameters and data ratios that produce the best performance for the classification of rice leaf diseases.

Here's a summary of the narrative per group of scenarios:

Scenarios 1–10 (90:10 Ratio) The model performs best at K=1 (Scenario 1) with only 4 total errors, all in Class 3. Performance decreases consistently as the value of K increases. Starting from K=4, Class 1 and Class 2 errors increase significantly. At K=10 (Scenario 10) the first error appears in the visual sample (True:1 → Pred:0).

Scenarios 11–20 (80:20 ratio) K=1 (Scenario 11) again show the best performance with minimal errors. A drastic decrease occurred starting from K=4, especially in Class 1 and Class 2. Starting from K=7 (Scenario 17) an error appears in the visual sample. Class 1 and Class 2 are consistently the classes that are most often misclassified from each other.

Scenario 21–30 (70:30 ratio) K=1 (Scenario 21) still excels with very few errors. However, in contrast to the previous ratio, errors in the visual sample have appeared since K=2 (Scenario 22). The number of errors jumped sharply at K=8–10, especially in Class 1 (up to 89 errors) and Class 2 (up to 101 errors).

Scenario 31–40 (60:40 ratio) The amount of test data is larger but the classification error is also greater in absolute terms. K=1 (Scenario 31) is still relatively best in this group. Scenarios 39–40 recorded the largest Class 1 and Class 2 errors of all scenarios (138 and 145 errors, respectively).

General conclusion: K=1 consistently provides the highest accuracy across all data ratios. Class 1 and Class 2 are the most vulnerable classes to misclassification, especially at large K values.

Analysis of Test Results

Table 1. Scenario results table

NO.	Splitting Data	Silhoutte Score	
		Training	Test
1	60:40	0,0738	0,1018
2	70:30	0,0644	0,0523
3	80:20	0,0563	0,0407
4	90:10	0,0507	0,0408

Silhouette Score is used as an evaluation metric in K-Means clustering to assess the extent to which a data is in the right cluster compared to other clusters. The value of this metric ranges from -1 to 1, where higher values indicate better clustering. In this study, testing was carried out with four scenarios for the distribution of training data and test data, namely 60:40, 70:30, 80:20, and 90:10. The results of the evaluation showed that the 60:40 scenario had the highest Silhouette Score on the training data of 0.0738, while the 90:10 scenario had the lowest value of 0.0507, indicating that the larger the portion of the training data, the quality of cluster separation did not necessarily improve due to the lack of variation in the data. In the test data, the 60:40 scenario again showed the best results with a Silhouette Score of 0.1018, while the 90:10 scenario had the lowest value of 0.0408, which suggests that too little test data can reduce the effectiveness of the model in forming clear clusters.

Although the 60:40 scenario has the highest Silhouette Score, the test results show that the model's accuracy rate is actually higher in the 90:10 scenario compared to other scenarios. This shows that although the clustering produced in the 90:10 scenario is less than optimal based on the Silhouette Score, a larger amount of training data is able to improve the model's performance in the rice leaf disease classification process using *K-Nearest Neighbor (KNN).* Therefore, in the selection of the data distribution ratio, it is necessary to consider the quality of the clustering and the level of accuracy of the model. If the main goal is to produce a more separate and well-structured cluster, then the 60:40 scenario becomes a more suitable option. However, if the primary focus is on improving classification accuracy, then a 90:10 scenario is more recommended because a larger amount of training data contributes to improved model performance in classifying new data.

Table 2. Scenario results table

METHODS	SPLIT DATES		Scenario	K	Accuracy
	TRAIN	TEST			
KMEANS & *K-Nearest Neighbor (KNN)*	90%	10%	1	1	88%
			2	2	86%
			3	3	86%
			4	4	82%
			5	5	84%
			6	6	81%
			7	7	85%
			8	8	84%
			9	9	84%
			10	10	82%
	80%	20%	11	1	48%
			12	2	44%
			13	3	44%
			14	4	36%
			15	5	38%
			16	6	34%
			17	7	39%
			18	8	35%

METHODS	SPLIT DATES		Scenario	K	Accuracy
	TRAIN	TEST			
			19	9	36%
			20	10	32%
	70%	30%	21	1	27%
			22	2	22%
			23	3	22%
			24	4	19%
			25	5	20%
			26	6	18%
			27	7	20%
			28	8	18%
			29	9	19%
			30	10	18%
	60%	40%	31	1	45%
			32	2	42%
			33	3	42%
			34	4	38%
			35	5	38%
			36	6	35%
			37	7	36%
			38	8	33%
			39	9	34%
			40	10	31%

Based on the table above, a test was carried out on a combination of K-Means and *K-Nearest Neighbor (KNN)* methods with various scenarios for sharing training data and test data, namely 90:10, 80:20, 70:30, and 60:40. In addition, variations were made to the K value to evaluate its effect on classification performance. Based on the test results, the 90:10 scenario yielded the highest accuracy, with an accuracy value range of 83% to 88% for various K-values.

Conversely, as the proportion of training data decreases and test data increases, there is a significant decrease in accuracy. This can be seen in the 60:40 scenario, where the accuracy ranges from 30% to 44%. This decrease indicates that with less trained data, the model becomes less able to recognize patterns in the data, thus reducing the classification ability of the test data. In addition, the test results also show that smaller K-values tend to result in higher accuracy than larger K-values. This is because a small K-value is more responsive to variations in data, while a larger K-value causes the model to become overgeneralized, which can ultimately reduce the accuracy of classification.

Based on the test results, it can be concluded that the selection of the right data ratio greatly affects the accuracy of the model, where the 90:10 ratio is the most optimal, as it provides the classification results with the highest accuracy. However, a smaller ratio like 60:40 leads to a pretty drastic drop in performance. In addition, the selection of K-values also has an important influence, where too large K-values can degrade accuracy because they cause the model to overconsider neighbors that may be irrelevant. Therefore, in the implementation of

the K-Means and *K-Nearest Neighbor (KNN)* methods, it is necessary to optimize the data sharing ratio and select the right K-value in order to obtain optimal classification results.

CONCLUSION

The results of this research demonstrate that the combination of K-Means clustering and K-Nearest Neighbor (KNN) classification can be effectively applied to classify rice leaf diseases using image segmentation techniques. The proposed model successfully identified four categories of rice leaf diseases, namely Bacterial Blight, Blast, Brown Spot, and Tungro, by utilizing preprocessing stages such as grayscale conversion, resizing, and feature transformation prior to segmentation and classification. The experimental results revealed that the highest classification accuracy reached 88% under the 90:10 training-testing data ratio, indicating that a larger proportion of training data significantly improves model performance. In addition, smaller K values generally produced higher classification accuracy compared to larger K values, showing that local neighborhood sensitivity plays an important role in disease recognition. Although several misclassifications still occurred between diseases with visually similar characteristics, the overall findings confirm that the integration of K-Means and KNN provides a practical and efficient approach for automatic rice leaf disease classification. This research also contributes to the development of image-based agricultural technology that can support early disease detection and improve decision-making in rice cultivation management. Despite the promising results, several limitations remain and open opportunities for future research development. The current study only focused on four rice leaf disease categories and relied primarily on grayscale image features, which may limit the model's ability to capture more complex texture and color information. Future research is therefore recommended to incorporate more advanced feature extraction methods, such as Gray Level Co-occurrence Matrix (GLCM), Local Binary Pattern (LBP), or deep learning-based feature representation to improve classification performance. In addition, future studies may compare the proposed K-Means and KNN combination with more sophisticated machine learning and deep learning algorithms such as Support Vector Machine (SVM), Random Forest, Convolutional Neural Network (CNN), or hybrid ensemble models. Expanding the dataset with more diverse environmental conditions, lighting variations, and additional disease categories is also important to improve model generalization and robustness. Furthermore, the development of a real-time mobile or web-based detection application would enhance the practical implementation of this system for farmers, agricultural extension workers, and researchers in supporting sustainable and technology-driven agriculture.

REFERENCES

Akanbi, O. N., Olawuyi, S. O., Adepoju, A. A., Olarinde, L. O., Dlamini, S. G., & Dlamini, D. V. (2024). Food security drive and adoption of improved rice varieties on the production efficiencies of upland and lowland rice farmers in north-central Nigeria. *Journal of Infrastructure, Policy and Development*, *8*(6), 3341.

Azzahra Nasution, D., Khotimah, H. H., & Chamidah, N. (2019). *PERBANDINGAN NORMALISASI DATA UNTUK KLASIFIKASI WINE MENGGUNAKAN ALGORITMA K-NN* (Vol. 4, Number 1).

Bhargava, A., Shukla, A., Goswami, O. P., Alsharif, M. H., Uthansakul, P., & Uthansakul, M. (2024). Plant leaf disease detection, classification, and diagnosis using computer vision and artificial intelligence: A review. *IEEE Access*, *12*, 37443–37469.

Harlina, P. W., Fitriansyah, F. A., & Shahzad, R. (2023). The challenging concept of diversifying non-rice products from cassava by changing Indonesian people's behavior and perception: a review. *Food Research*, *7*(5), 251–259.

Jafar, A., Bibi, N., Naqvi, R. A., Sadeghi-Niaraki, A., & Jeong, D. (2024). Revolutionizing agriculture with artificial intelligence: plant disease detection methods, applications, and their limitations. *Frontiers in Plant Science*, *15*, 1356260.

Jin, T., & Zhong, T. (2022). Changing rice cropping patterns and their impact on food security in southern China. *Food Security*, *14*(4), 907–917.

Khan, S. U., Alsuhaibani, A., Alabduljabbar, A., Almarshad, F., Altherwy, Y. N., & Akram, T. (2025). A review on automated plant disease detection: motivation, limitations, challenges, and recent advancements for future research. *Journal of King Saud University Computer and Information Sciences*, *37*(3), 34.

Li, C., & Wang, M. (2024). Pest and disease management in agricultural production with artificial intelligence: Innovative applications and development trends. *Advances in Resources Research*, *4*(3), 381–401.

Mohidem, N. A., Hashim, N., Shamsudin, R., & Che Man, H. (2022). Rice for food security: Revisiting its production, diversity, rice milling process and nutrient content. *Agriculture*, *12*(6), 741.

Negi, P., & Anand, S. (2024). Plant disease detection, diagnosis, and management: Recent advances and future perspectives. *Artificial Intelligence and Smart Agriculture: Technology and Applications*, 413–436.

Nizamani, M. M., Zhang, Q., Muhae-Ud-Din, G., & Wang, Y. (2023). High-throughput sequencing in plant disease management: a comprehensive review of benefits, challenges, and future perspectives. *Phytopathology Research*, *5*(1), 44.

Premana, A., & Pandhu Wijaya, A. (2022). *Klasifikasi Jenis Buah Mangga Menggunakan Metode K-Means Clustering*. *5*.

Rekayasa, K. K., Khoiruddin, M., Junaidi, A., & Saputra, W. A. (2022). Journal of Dinda Klasifikasi Penyakit Daun Padi Menggunakan Convolutional Neural Network. *Data Institut Teknologi Telkom Purwokerto*, *2*(1), 37–45. https://www.kaggle.com/tedisetiady/leaf-rice-disease-

Rozi, F., Santoso, A. B., Mahendri, I. G. A. P., Hutapea, R. T. P., Wamaer, D., Siagian, V., Elisabeth, D. A. A., Sugiono, S., Handoko, H., & Subagio, H. (2023). Indonesian market demand patterns for food commodity sources of carbohydrates in facing the global food crisis. *Heliyon*, *9*(6).

Sitaresmi, T., Hairmansis, A., Widyastuti, Y., Susanto, U., Wibowo, B. P., Widiastuti, M. L., Rumanti, I. A., Suwarno, W. B., & Nugraha, Y. (2023). Advances in the development of rice varieties with better nutritional quality in Indonesia. *Journal of Agriculture and Food Research*, *12*, 100602.

Suban, I. B., Paramartha, A., Fortwonatus, M., & Santoso, A. J. (2020). Identification the Maturity Level of Carica Papaya Using the K-Nearest Neighbor. *Journal of Physics: Conference Series*, *1577*(1). https://doi.org/10.1088/1742-6596/1577/1/012028

Sutardi, C., Apriyana, Y., Rejekiningrum, P., Alifia, A. D., Ramadhani, F., Darwis, V., Setyowati, N., Setyono, D. E. D., Gunawan, & Malik, A. (2022). The transformation of rice crop technology in Indonesia: innovation and sustainable food security. *Agronomy*, *13*(1), 1.

Tasfe, M., Nivrito, A. K. M., Al Machot, F., Ullah, M., & Ullah, H. (2024). Deep learning based models for paddy disease identification and classification: A systematic survey. *IEEE Access*, *12*, 100862–100891.

Trisnawan, A., Harianto, W., Informatika, T., Sains, F., Universitas, D. T., & Malang, K. (2019). Klasifikasi Beras Menggunakan Metode K-Means Clustering Berbasis Pengolahan Citra Digital. In *Jurnal Terapan Sains & Teknologi (RAINSTEK)* | (Vol. 1, Number 1).

Vikram, N., & Sarkar, N. C. (2022). Rice Bean (Vigna umbellata) – A Potential Legume under Adverse Conditions. *International Journal of Bio-Resource and Stress Management*, *13*(9), 928–934. https://doi.org/10.23910/1.2022.3117a

Widyanti, T., Hilabi, S. S., Hananto, A., Tukino, & Novalia, E. (2023). Implementasi K-Means dan K-Nearest Neighbors pada Kategori Siswa Berprestasi. *Jurnal Informasi Dan Teknologi*, *5*(1), 75–82.

Worung, D. T., Sompie, S. R. U. A., & Jacobus, A. (2020). Implementasi K-Means dan K-NN pada Pengklasifikasian Citra Bunga. *Jurnal Teknik Informatika*, *15*(3), 217–222.