

# HOG Feature Extraction in Optimizing FK-NN and CNN for Image Identification of Rice Plant Diseases

Adie Wahyudi Oktavia Gama<sup>1,\*</sup>, Putu Vina Junia Antarista Gunawan<sup>2</sup>, Kadek Darmaastawan<sup>3</sup>

<sup>1,2,3</sup>Department of Information Technology, Faculty of Engineering and Informatics, Universitas Pendidikan Nasional, Denpasar, Bali 80224, Indonesia.

(Received: February 10, 2025; Revised: March 20, 2025; Accepted: May 1, 2025; Available online: June 15, 2025)

## Abstract

This study compares the performance of FK-NN and CNN models in identifying rice diseases from digital images, focusing on both effectiveness and efficiency. Additionally, this research utilizes HOG for feature extraction from the digital images. The stages include data collection, preprocessing, transformation, classification, and model evaluation. The results show that the FK-NN model achieves a higher accuracy of 86.26%, compared to the CNN model's accuracy of 71.37%. Furthermore, the precision value of the FK-NN model is also higher at 86.88%, compared to the CNN model's precision of 72.74%. Similarly, the recall value for the FK-NN model is higher at 86.88%, compared to the CNN model's 71.37%. The F1-score of the FK-NN model is likewise superior, with a value of 86.88%, compared to the CNN model's F1-score of 71.37%. These findings suggest that the FK-NN model with HOG feature extraction is more effective. However, in terms of inference time, the CNN model is faster, taking 0.000282 seconds compared to FK-NN's 0.002331 seconds. In conclusion, the FK-NN model with HOG feature extraction excels in identifying rice diseases, while the CNN model offers faster inference time in this study.

*Keywords:* Rice Disease, FK-NN, CNN, HOG Feature Extraction, Machine Learning

## 1. Introduction

In Indonesia, the dependence on rice as the primary source of carbohydrates is very high, with more than 90% of the population's carbohydrate consumption relying on rice as a staple food [1], [2]. According to data from the Central Statistics Agency in 2023, the area of rice fields in Indonesia is recorded at 10.20 million hectares, with a production of 53.63 million tons of milled dry unhusked rice, although there has been a 2.45% decrease in harvested area compared to data from 2022 [3]. One of the main factors contributing to this decline is disease attacks on rice plants, which cause significant losses, with an estimated 200,000 to 300,000 tons of crop failure every year [4], [5], [6], [7]. Therefore, rapid, precise, and accurate disease identification measures are necessary to prevent further losses and ensure food production stability at the national level. Conventional methods often used by farmers to identify diseases in rice plants tend to be neither fast nor accurate enough. This is due to the limited knowledge farmers have in detecting diseases in a timely manner, which often leads to delays or mistakes in handling the issue. Therefore, more efficient solutions are needed for detecting rice plant diseases.

An effective strategy involves leveraging machine learning for image analysis to detect plant diseases [8], [9]. This approach facilitates rapid and precise disease identification, allowing for timely intervention to minimize potential losses [10]. The evolution of machine learning is closely linked to the remarkable advancements in information and communication technology in recent decades. These developments have also spurred the integration of artificial intelligence (AI) across various industries, including agriculture. As a subset of AI, machine learning empowers systems to learn from data and generate predictions or decisions based on that learning. In the agricultural sector, machine learning significantly enhances decision-making efficiency by delivering automated predictive insights, thereby improving the effectiveness and productivity of plant disease management [11].

\*Corresponding author: Adie Wahyudi Oktavia Gama (adiewahyudi@undiknas.ac.id)

DOI: <https://doi.org/10.47738/jads.v6i3.722>

This is an open access article under the CC-BY license (<https://creativecommons.org/licenses/by/4.0/>).

© Authors retain all copyrights

Several machine learning algorithms are available for agricultural disease detection, including the Fuzzy K-Nearest Neighbor (FK-NN) method [12]. The FK-NN algorithm excels at managing uncertain and noisy data, delivering improved accuracy in scenarios where data clarity is compromised [13], [14], [15]. In contrast, algorithms like Convolutional Neural Networks (CNN) have demonstrated exceptional effectiveness in addressing image classification challenges by automating feature extraction, which enhances their performance in digital image analysis [16], [17], [18]. Previous studies indicate that both FK-NN and CNN algorithms have consistently achieved strong accuracy in various research applications.

Therefore, this study aims to compare the performance of these two machine learning models in identifying digital images of diseases in rice plants. Additionally, this research uses the Histogram of Oriented Gradients (HOG) method for feature extraction from digital images. The HOG method allows for the extraction of texture and shape features from rice leaf images, which helps improve accuracy in disease identification through both models [19], [20], [21]. Thus, this study is expected to make a significant contribution to the development of fast and accurate disease detection models for rice plants, supporting efforts to achieve national food security. Specifically, this research aims to conduct a comparative analysis between the FK-NN and CNN algorithms in detecting rice disease images by utilizing HOG features in both algorithms.

## 2. Method

Figure 1 shows the overall framework used in the research comparing the FK-NN and CNN methods for the digital image identification of rice diseases using HOG feature extraction.

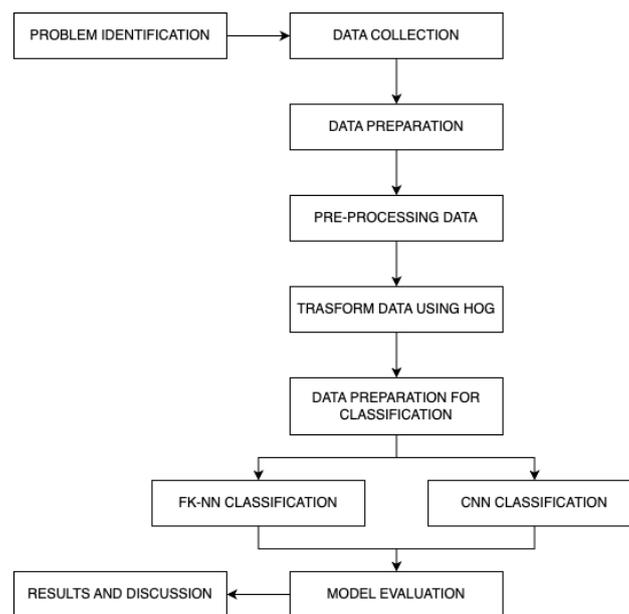


Figure 1. Research overview

### 2.1. Data Collection

The research utilizes a comprehensive collection of digital images featuring rice leaves in various conditions, including both healthy specimens and leaves exhibiting symptoms of common rice diseases. These images were carefully captured under controlled lighting conditions to ensure consistency in quality and resolution. The collected dataset will be strategically divided into training and testing subsets to develop and validate the machine learning models for accurate disease detection and classification.

### 2.2. Data Pre-processing

The data pre-processing stage is a crucial step that prepares the raw dataset for effective use in both training and testing phases of the machine learning pipeline. In this research, the pre-processing involves several key operations: converting color images to grayscale to simplify analysis, resizing all images to a uniform dimension for consistency, and carefully labeling each image to indicate whether it shows healthy or diseased rice leaves.

### 2.3. Data Transformation

Data transformation is an advanced step in pre-processing, where the processed data is transformed into a form more suitable for modeling. In this phase, feature extraction is carried out using the Histogram of Oriented Gradients (HOG) technique. HOG is a method that utilizes gradient intensity to extract local features from an object, which are then used in machine learning and image processing [22], [23].

The process of extracting HOG features from digital images begins with image pre-processing, where the image is converted from color to grayscale to simplify subsequent steps. To ensure robustness against lighting variations, pixel intensity is normalized using techniques such as gamma correction or equalization. This adjustment helps mitigate the impact of uneven illumination, providing a more consistent foundation for feature extraction. Next, the image is divided into small regions called cells, typically measuring 8x8 pixels. This division allows the algorithm to focus on localized gradient information, which is crucial for identifying edges and textures within the image. By analyzing these smaller regions, the method captures fine details that contribute to the overall structure of objects. Following this, gradients are calculated to determine both the magnitude and orientation of intensity changes across the image. The gradient magnitude reflects the strength of these changes, while the orientation indicates their direction. These gradients are fundamental to HOG, as they highlight key structural elements such as edges and corners, forming the basis for further analysis. Once gradients are computed, a histogram of gradient orientations is constructed for each cell. The orientation range is divided into nine bins spanning 0 to 180 degrees, a carefully chosen balance that preserves essential directional information without unnecessary complexity. The gradient magnitudes are accumulated into these bins, resulting in a histogram that represents the distribution of gradient directions within the cell. To enhance feature robustness, multiple cells are grouped into overlapping blocks, typically consisting of 2x2 cells. The histograms within each block are concatenated and normalized using methods such as L2-norm or L1-norm. This normalization reduces sensitivity to local contrast variations, ensuring that the features remain stable under different lighting conditions. Finally, the normalized histograms from all blocks are combined into a single feature vector, which encapsulates the gradient structure of the entire image. This comprehensive representation is highly effective for tasks such as object detection and classification, as it efficiently encodes the essential shape and texture information needed for accurate recognition.

In this study, the data transformation process involves feature extraction using HOG, transforming the data into a feature vector, and normalizing the features. Comparative analyses demonstrate that the HOG feature extractor performs better than other feature extraction techniques, such as SIFT, SURF, and HAAR, when applied to imbalanced datasets [24]. Therefore, HOG was selected for its efficiency and effectiveness in capturing essential features for object detection tasks, especially when dealing with image representations that focus on gradient orientations rather than precise key points.

### 2.4. Classification using Fuzzy K-Nearest Neighbors (FK-NN)

FK-NN is a classification method that combines fuzzy logic with the conventional K-NN algorithm [25], [26], [27]. Unlike traditional K-NN, which assigns a data point to a single class based on a majority vote of its nearest neighbors, FK-NN introduces the concept of fuzzy membership. This allows data points to belong to multiple classes simultaneously, with varying degrees of membership in the interval [0,1]. This flexibility is particularly useful in handling ambiguous or overlapping data, where a clear-cut classification may not be possible. The FK-NN classification process begins by calculating the Euclidean distance between feature vectors, which measures the straight-line distance between two points in feature space. This distance is computed using the formula:

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2} \quad (1)$$

$d(x_i, x_j)$  = Euclidean distance;  $x_i$  = Feature vector of i-th data;  $x_j$  = Feature vector of j-th data;  $x_{ik}$  = Value of the k-th feature from the feature vector  $x_i$ ;  $x_{jk}$  = Value of the k-th feature from the feature vector  $x_j$ ;  $n$  = Number of features in the feature vector.

Following this distance calculation, the algorithm determines the fuzzy membership degree through an exponential function that incorporates both the computed Euclidean distance and a standard deviation parameter. This membership degree, expressed as:

$$\mu_{ci}(x) = \exp\left(-\frac{d(x,ci)^2}{2\sigma^2}\right) \quad (2)$$

$\mu_{ci}(x)$  = Membership degree;  $\exp$  = Exponential function;  $d(x, ci)^2$  = Euclidean distance;  $\sigma$  = Standard deviation parameter.

The fuzzy membership degree reflects the uncertainty in class assignments. A higher membership value indicates a stronger association with a particular class, while lower values suggest weaker or partial membership. The classification process then employs a fuzzy weighting method that evaluates each test data point's membership degree across all classes, considering only its K nearest neighbors. Unlike conventional K-NN where neighbors contribute equally, FK-NN uses fuzzy logic to weigh each neighbor's influence based on proximity and membership degree. This approach proves particularly valuable for handling ambiguous or overlapping data, as it moves beyond binary decisions to consider nuanced class belongingness. The final class prediction emerges through an aggregation process that selects the class with the highest average membership degree among the K nearest neighbors, mathematically represented as:

$$\text{Class}(x) = \arg \max_{c_i} \left( \frac{1}{K} \sum_{j=1}^K \mu_{c_i}(x_j) \right) \quad (3)$$

$\text{Class}(x)$  = The final class predicted for the test data  $x$ ;  $c_i$  = The  $i$ -th class;  $K$  = The number of nearest neighbors considered;  $\mu_{c_i}(x_j)$  = The fuzzy membership degree.

Fuzzy weighting is applied to account for the influence of distance, where closer neighbors have a greater impact on the classification result. This weighting can be determined based on the fuzzy membership function. In conventional K-NN, each neighbor contributes equally to the classification decision, regardless of their distance from the test point. In contrast, FK-NN incorporates fuzzy logic to weigh the influence of each neighbor based on their proximity and membership degree. This allows FK-NN to handle overlapping or ambiguous data more effectively, as it considers the degree of belongingness to each class rather than making a binary decision. In this study, for the FK-NN model, the parameter determination is done by setting the value of K and the fuzzy membership function. The FK-NN model with its parameters set is then trained using the training data and makes predictions using the testing data to identify rice diseases.

## 2.5. Classification using Convolution Neural Network (CNN)

CNN consists of two main architectural layers: feature learning and classification layer. In the feature learning layer, there are the input layer, convolution layer, activation layer, and pooling layer, while in the classification layer, there are the fully connected layer and output layer [28], [29], [30], [31]. At the heart of CNNs is the convolution operation, which extracts essential features like edges, textures, and patterns from images. Imagine a small filter (or kernel) sliding across the image, much like a magnifying glass moving over a photograph. At each step, the filter multiplies its values with the corresponding pixel values of the image beneath it. These products are then summed up to create a single value, which forms part of the output feature map. This process repeats as the filter moves across the entire image, capturing local patterns that help the network understand the image's structure. The formula represented as:

$$(I * K)(i, j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(i + m, j + n) \times K(m, n) \quad (4)$$

$I$  = Input matrix (image);  $K$  = Kernel;  $M$  = Kernel dimension (high);  $N$  = Kernel dimension (width);  $(i, j)$  = Coordinates on the resulting feature map.

After convolution, the ReLU (Rectified Linear Unit) activation function is applied. ReLU simply takes the output of the convolution and sets all negative values to zero while keeping positive values unchanged. This introduces non-linearity into the model, allowing it to learn complex patterns. ReLU is widely preferred because it is computationally efficient and helps avoid issues like the vanishing gradient problem, which can hinder training in deeper networks. Studies have shown that ReLU outperforms other activation functions, such as Sigmoid and Tanh, in tasks like image classification [32], [33]. This operation, mathematically expressed as:

$$f(x) = \max(0, x) \quad (5)$$

$x$  = Input value.

The pooling layer reduces the size of the feature maps, making the model more efficient and less prone to overfitting. Two common types of pooling are max pooling and average pooling. Max pooling selects the highest value from a region of the feature map, while average pooling calculates the average value. Both methods help retain the most important features while reducing the computational load.

$$\text{Max: } O(i, j) = \max\{I(m, n) \mid m \in [i \times S, i \times S + F - 1], n \in [j \times S, j \times S + F - 1]\} \quad (6)$$

$$\text{Average: } O(i, j) = \frac{1}{F^2} \sum_{m=iS}^{iS+F-1} \sum_{n=jS}^{jS+F-1} I(m, n) \quad (7)$$

I = Input matrix; O = Output of pooling; S = Stride; F = Size of the pooling filter.

Finally, the fully connected layer takes the features extracted by the previous layers and combines them to produce the final output. This layer uses weights and biases to make predictions based on the learned features.

$$z = W \times x + b \quad (8)$$

W = Weight matrix; x = Input; b = Bias.

In this study, the CNN model involves determining the CNN architecture, including the number of layers, types of layers, and activation functions. The CNN model is then trained using training data and tested using testing data to identify rice diseases.

## 2.6. Model Evaluation

The evaluation in this study will use a confusion matrix to assess the performance of each model. The evaluation results of FK-NN and CNN are compared to determine which method is more effective in identifying rice diseases.

$$F1 - \text{Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (9)$$

$$\text{Accuracy} = \frac{\text{Num\_of\_Correct\_Prediction}}{\text{TotalPrediction}} \quad (10)$$

$$\text{Precision} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalsePositive}} \quad (11)$$

$$\text{Recall} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalseNegative}} \quad (12)$$

In addition to evaluation using the confusion matrix, inference time will also be measured. Inference time is the amount of time required by the machine learning model to process new input data and generate predictions or outputs after the model has been trained. Inference time is an important metric in evaluating model performance, especially in applications that require fast and real-time predictions.

## 3. Results and Discussion

This study uses Google Interactive Notebook (Google Colab) as a supporting tool for building machine learning models. Google Colab is equipped with various libraries such as Keras, TensorFlow, NumPy, Pandas, and other tools. For model training, Google Colab's free tier GPU resources were utilized, which typically include NVIDIA Tesla K80 or T4 GPUs. These GPUs provide significant computational power, enabling efficient training of deep learning models like CNNs. However, the availability of these resources can vary depending on usage demand, which may occasionally impact training times. The use of GPU acceleration was particularly beneficial for handling the computationally intensive tasks involved in training and evaluating the FK-NN and CNN models.

### 3.1. Data Collection

The data collection phase in this study begins by identifying digital images related to rice plant diseases available on Kaggle.com (<https://www.kaggle.com/datasets/imbikramsaha/paddy-doctor>) uploaded by Bikram Saha. This dataset contains a folder called train\_images, which includes 10 subfolders with nine different rice plant diseases and one folder containing healthy rice plants. The dataset contains 10,407 images in .jpg format as illustrated in table 1.

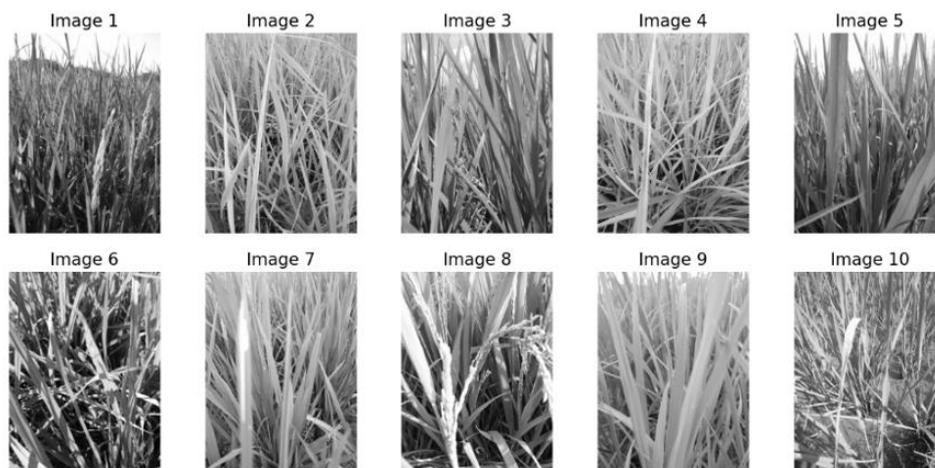
**Table 1.** Dataset structure

No	Folder Name	Files Number
1	BacterialLeafBlight	479 files
2	BacterialLeafStrek	380 files
3	BacterialPanicleBlight	337 files
4	Blast	1.738 files
5	BrownSpot	965 files
6	DeadHeart	1.442 files
7	DownyMildew	620 files
8	Hispa	1.594 files
9	Normal	1.764 files
10	Tungro	1.088 files

Table 1 shows the details of the dataset structure used in this study. Each folder in the dataset is organized based on the type of disease, which facilitates the annotation and analysis process. All data in the dataset will be used in both the training and testing of the model, without any special treatment to address data imbalance.

### 3.2. Data Pre-processing

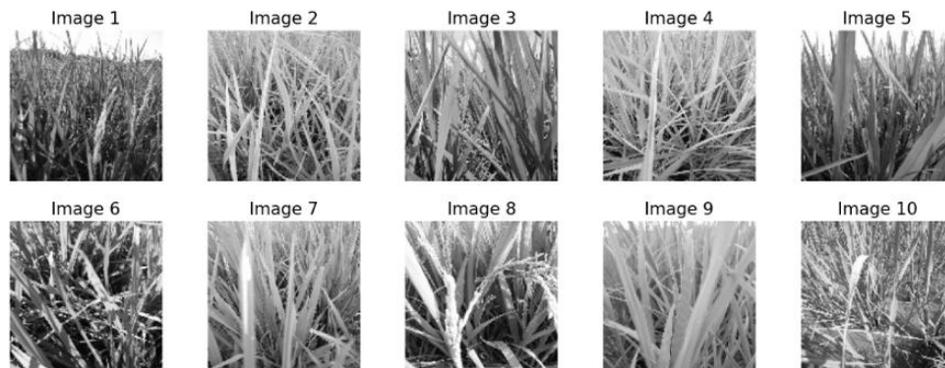
The data will undergo preprocessing steps, which include converting the data to grayscale, resizing the data, and labeling the data. The first step in preprocessing is converting the color images to grayscale format. In this phase, each image loaded from the dataset folder is transformed into grayscale. This conversion is performed to simplify the visual information, reduce data complexity, and ensure consistent image format. Converting images to grayscale is important for the next feature extraction stage using the HOG method, which focuses on detecting gradients and edge orientations within the image. All images are converted to grayscale to eliminate irrelevant visual elements, such as color, which could affect the training process, and to make the HOG feature extraction more effective. The result of this conversion process can be seen in figure 2.



**Figure 2.** Data conversion results

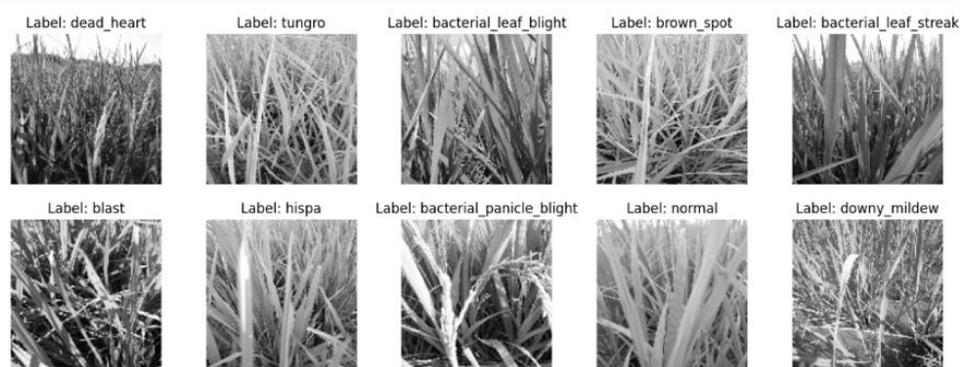
The next step in preprocessing involves resizing the images to ensure all data points in the dataset have the same dimensions. For this study, the images were resized to 128x128 pixels. This standardization simplifies the training process and speeds up computations while preserving essential visual details. The choice of 128x128 pixels was carefully considered to strike a balance between maintaining image quality and ensuring computational efficiency. This size is large enough to capture important features, such as the patterns and textures on rice leaves, which are critical for accurate analysis, as illustrated in figure 3. At the same time, it is small enough to keep the computational load

manageable. Through experimentation, it was found that smaller sizes often led to the loss of key details, while larger sizes increased processing time without a noticeable improvement in accuracy. As a result, 128x128 pixels emerged as the optimal resolution for this dataset, enabling both efficient training and reliable feature extraction.



**Figure 3.** Resize data results

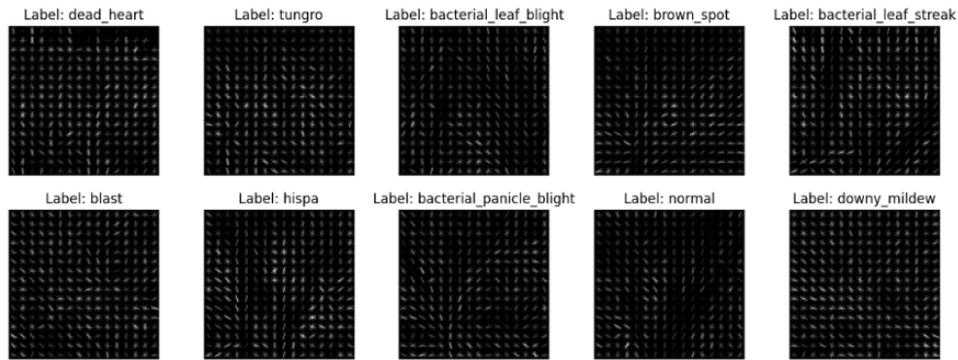
The final step in the preprocessing process is data labeling. In this phase, each processed image is given a label according to the rice disease category. The labeling is done automatically based on the folder name where the image is stored. This process ensures that each data point has an accurate and consistent label, which will be used as the target during the model training process. To verify the accuracy of the labeling, manual checks are performed periodically to ensure that the labels are correctly assigned to the images. Data labeling, as depicted in figure 4 below, is a crucial step in the development of supervised learning-based machine learning models because these labels serve as a reference for the model to learn patterns corresponding to each disease category.



**Figure 4.** Data labeling results

### 3.3. Data Transformation

The data transformation process involves feature extraction using HOG, transforming data into feature vectors, and normalizing features. The first step in the data transformation process is feature extraction using HOG. HOG feature extraction aims to identify relevant visual patterns by detecting gradient directions around the pixels in an image. This process generates a data representation in the form of a histogram that depicts gradient orientations for each part of the image (see figure 5).



**Figure 5.** Feature extraction results using HOG

Once the features are extracted, the data is transformed into feature vectors. This transformation organizes the information from each image in the dataset into a uniform vector format. This vector format simplifies the computational process and analysis in the machine learning model, as the data becomes more structured and ready for efficient processing. In this phase, all feature vectors extracted from the HOG process are combined into a feature matrix. This matrix contains the vector representation of all the data in the dataset, which will later be used as input for the machine learning model training process, as pictured in [figure 6](#) below.

```

Feature Vectors:
[[0.25583564 0.0381047 0.02038166 ... 0.05072051 0.14858325 0.08370363]
 [0.06915587 0.01458145 0.02511069 ... 0.04463701 0.02273906 0.03814898]
 [0.16035245 0.03337164 0.14584595 ... 0.07272148 0.05749806 0.07898779]
 ...
 [0.02564196 0.003736 0.00752137 ... 0.11164772 0.10378519 0.21580404]
 [0.12785371 0.00067894 0.00969497 ... 0.14485611 0.03241863 0.13329469]
 [0.13940179 0.07442479 0.00383791 ... 0.08387029 0.01919565 0.09489003]]
    
```

**Figure 6.** Feature vector transformation results

The final step in the data transformation process is feature normalization. In this step, the values within the feature vector are adjusted to lie within a certain range, such as between zero and one or with a standard deviation of one. Feature normalization is done to reduce scale differences between various features that may have very different value ranges, as illustrated in [figure 7](#) below.

```

Normalized Features:
[[ 2.14892831 -0.13061299 -0.45747637 ... -0.34752575 1.63234461
  0.48829804]
 [-1.03029288 -0.71246011 -0.34383108 ... -0.43473296 -0.67683299
 -0.38877947]
 [ 0.52281653 -0.24768525 2.55760372 ... -0.03214081 -0.03902278
 0.3975025 ]
 ...
 [-1.77134988 -0.98072196 -0.7665268 ... 0.52586862 0.81032275
 3.03166723]
 [-0.03064841 -1.05633839 -0.7142923 ... 1.00191252 -0.49921774
 1.44308972]
 [ 0.16601949 0.76776336 -0.85504554 ... 0.12767789 -0.74185275
 0.70367321]]
    
```

**Figure 7.** Feature normalization results

### 3.4. Classification Data Preparation

Before moving on to the classification process, the transformed data will go through a label encoding stage. This is necessary because the labels in the original dataset are in text or categorical form, which cannot be directly understood by machine learning models. Therefore, categorical labels are converted into numeric values so they can be recognized and processed by the algorithm, as described in [figure 8](#).

```
Label and encoding:  
bacterial_panicle_blight: 2  
bacterial_leaf_streak: 1  
hispa: 7  
bacterial_leaf_blight: 0  
downy_mildew: 6  
brown_spot: 4  
dead_heart :5  
tungro: 9  
normal: 8  
blast:3
```

**Figure 8.** Label encoding results

After the encoding process is complete, the next step is data splitting into two parts: training data and testing data. This phase is important for evaluating the model's performance. By separating training and testing data, the model can be trained to learn patterns from the training data, while the testing data is used to assess the model's ability to predict unseen data. In this study, an 80:20 ratio was used. This ratio is a common choice in machine learning, as it provides a sufficient amount of data for training while retaining a meaningful portion for evaluation. The 80:20 split ensures that the model has enough data to learn robust patterns without overfitting, while the 20% testing set allows for a reliable assessment of generalization performance. Based on these calculations, the training data is 8325 and the testing data is 2082.

### 3.5. FK-NN Classification

FK-NN classification involves determining FK-NN parameters, training the FK-NN and testing the FK-NN model. The initial step in FK-NN classification is determining the parameters to be used by the model. In the FK-NN method, two main parameters need to be determined: the number of nearest neighbors ( $K$ ) and the fuzzy parameter ( $m$ ). The  $K$  parameter determines how many nearest neighbors will be considered when making classification decisions, while the  $m$  parameter controls the degree of membership weighting in the fuzzy function.

Parameter tuning is done to find the optimal combination of  $K$  and  $m$  that yields the best results for rice disease classification. This tuning is done using 20% of the data as training data, and various parameter values are tested:  $K = 3, 5, 7, 9$  and  $m = 1.5, 2, 2.5, 3$ . The results of parameter tuning show that the best combination is  $K = 3$  and  $m = 3$ . Once the parameters are determined, these optimal values of  $K$  and  $m$  will be used for training the FK-NN model for rice disease classification.

The next step in FK-NN classification is training the model using the parameters determined earlier. This training process is done with training data that has gone through various preprocessing and transformation stages. The processed data will be used to teach the model about the patterns in the dataset, so that it can more accurately recognize and distinguish between different types of rice diseases. The FK-NN model training process uses  $K = 3$  and  $m = 3$ , as obtained from the parameter tuning stage.

After the FK-NN model is trained, the next step is making predictions on new rice disease images that were not used during training. This prediction process uses test data to evaluate how well the trained model can classify unseen data. The prediction results will show the rice disease class based on the input image and will be used in model evaluation. [Figure 9](#) shows five example prediction results from the FK-NN model, where the model successfully identified all classes correctly according to their actual values.

```
Prediction: [4 3 8 8 1]  
Actual: [4 3 8 8 1]
```

**Figure 9.** FK-NN model prediction results

Class 4 representing brown\_spot, the model was able to predict accurately without error. Similarly, for class 3 (blast), the model demonstrated excellent performance by recognizing the pattern correctly. Moreover, for class 8 (normal), even though there were two instances in the test dataset, both were classified correctly, confirming the model's ability

to differentiate data from this class. The same result was observed for class 1 (bacterial\_leaf\_streak), where the model gave predictions consistent with the true values.

### 3.6. CNN Classification

CNN classification involves determining the CNN architecture, training the CNN model, and testing the CNN model. The first step in implementing CNN is determining the network architecture to be used. In this study, CNN is used to perform classification based on HOG feature extraction input. The CNN architecture implemented in this research is designed to process data that has been converted from HOG feature vectors into a 3D matrix form, so it can be further processed by the convolutional layers.

The CNN model in this study uses several key layers, including reshape, Conv2D, MaxPooling2D, and Dense layers. The reshape layer is used to adjust the input data dimensions to fit the desired shape of the network. Then, the Conv2D layer is used to extract important features from the images through convolution operations. Next, the MaxPooling2D layer is applied to reduce data dimensions and decrease computational complexity, while the Dense layer is used for the final classification based on the features extracted.

Once the CNN architecture is determined, the next step is training the model using the training data that has gone through preprocessing and transformation. During training, the processed data will be fed into the network for several epochs. In this study, the CNN model is trained for 20 epochs, which allows the model to iteratively update its weights to improve prediction accuracy. After the CNN model is trained, the next step is making predictions on new rice disease images. At this stage, the trained model will be tested using data that was not used during training. The goal of the prediction is to classify the input image into the appropriate rice disease class. [Figure 10](#) shows five example prediction results from the CNN model on test data, where the model produced varying accuracy levels compared to the actual values.

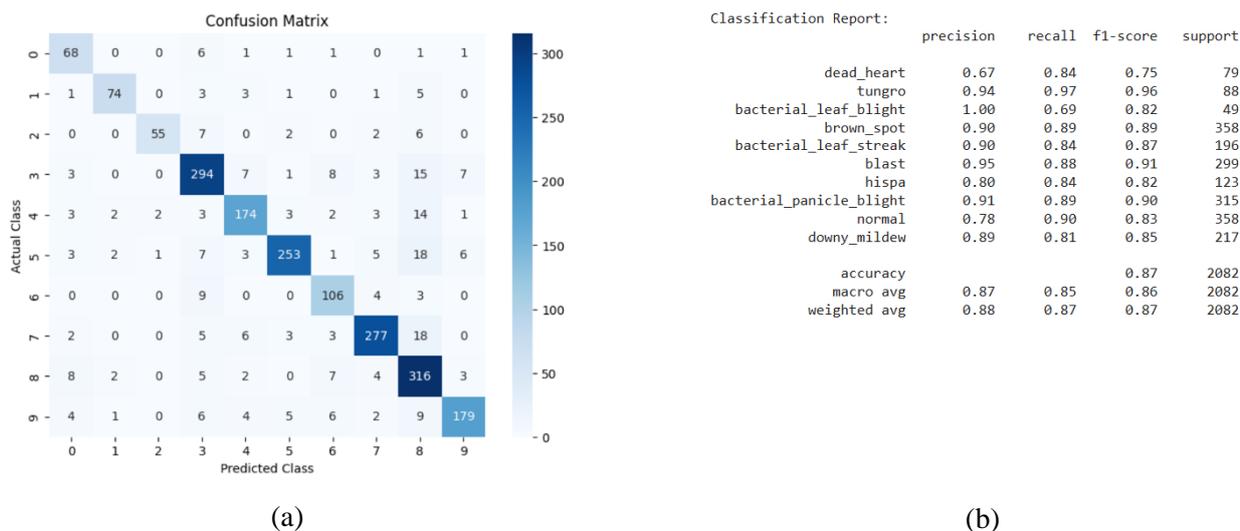
```
Prediction: [4 3 8 8 1]  
Actual: [4 3 8 8 1]
```

**Figure 10.** CNN model prediction results

Class 3 (blast), the model correctly predicted the first two instances with full accuracy, matching the original labels. However, there was an error for class 7 (hispa), where one instance was predicted as class 8 (normal), indicating that the model struggled to differentiate between these two classes. Furthermore, for class 8 (normal), the model predicted two instances correctly, although one actually came from class 3 (blast).

### 3.7. Model Evaluation Result

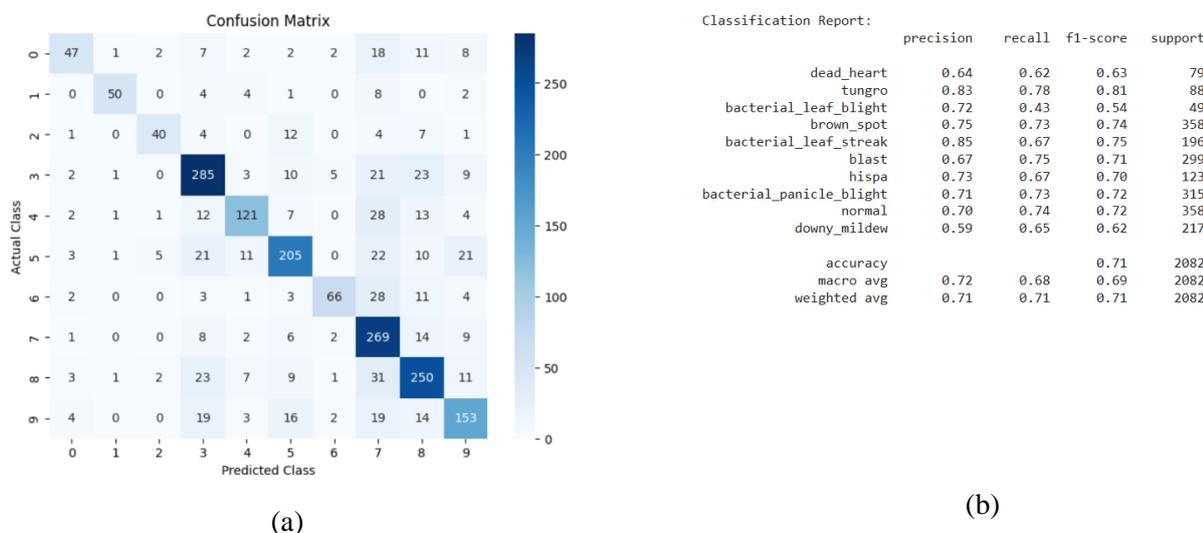
The model evaluation in this study used a confusion matrix, which is used to calculate accuracy, precision, recall, and F1-score. Additionally, inference time measurements were made to evaluate the time required by the model to make predictions on test data. The FK-NN model confusion matrix with HOG feature extraction results is shown in [figure 11](#).



**Figure 11.** (a) Confusion matrix FK-NN (b) Classification report

Based on the confusion matrix presented in figure 11, essential performance indicators such as accuracy, precision, recall, and F1-score can be derived to assess the effectiveness of the FK-NN model. In the matrix, Class 1 corresponds to bacterial\_leaf\_streak, Class 0 to bacterial\_leaf\_blight, Class 2 to bacterial\_panicle\_blight, Class 3 to blast, Class 4 to brown\_spot, Class 5 to dead\_heart, Class 6 to downy\_mildew, Class 7 to hispa, Class 8 to normal, and Class 9 to tungro. The model demonstrated an accuracy of 86.26%, indicating the percentage of correct predictions out of the total predictions. Additionally, it achieved a precision of 86.88%, highlighting its ability to make accurate positive predictions.

The confusion matrix reveals that the FK-NN model occasionally misclassifies certain diseases, such as bacterial\_leaf\_blight (Class 0) and blast (Class 3), which are sometimes confused with brown\_spot (Class 4). This could be due to similarities in visual symptoms, such as leaf discoloration or lesions, making it challenging for the model to distinguish between them. Additionally, false negatives for tungro (Class 9) suggest that the model may struggle with detecting this disease, possibly because of its subtle or less distinct visual features compared to others. These insights highlight areas where the model could be improved, such as by incorporating additional features or enhancing the training data for these specific classes. The confusion matrix for the FK-NN model with HOG feature extraction is shown in figure 12.



**Figure 12.** (a) Confusion matrix CNN (b) Classification report

The confusion matrix in [figure 12](#), provides the basis for computing critical performance measures, including accuracy, precision, recall, and F1-score, which collectively evaluate the CNN model's effectiveness. In the matrix, Class 1 denotes bacterial\_leaf\_streak, Class 0 stands for bacterial\_leaf\_blight, Class 2 for bacterial\_panicle\_blight, Class 3 for blast, Class 4 for brown\_spot, Class 5 for dead\_heart, Class 6 for downy\_mildew, Class 7 for hispa, Class 8 for normal, and Class 9 for tungro. The model achieved an accuracy of 71.37%, reflecting the ratio of correct predictions to the total predictions. Its precision of 72.74% demonstrates its capability to correctly identify positive instances, while a recall of 71.37% reveals its effectiveness in capturing all actual positives. The F1-score, at 71.20%, represents a balanced measure of precision and recall. Additionally, the FK-NN model's inference time is 0.000282 S per sample.

The CNN model shows higher misclassification rates for bacterial\_panicle\_blight (Class 2) and tungro (Class 9), often confusing them with bacterial\_leaf\_blight (Class 0) and normal (Class 8), respectively. This could be attributed to the overlapping visual characteristics of these diseases or insufficient representation of these classes in the training data. For example, tungro's symptoms might resemble healthy leaves in certain stages, leading to false negatives. Addressing these issues could involve augmenting the dataset with more diverse examples or fine-tuning the model to better capture subtle differences between classes.

### 3.8. Discussions

A comparative analysis is performed between the two models used in the study, namely FK-NN and CNN. This analysis aims to evaluate the performance of each model in the context of rice disease classification, based on various evaluation metrics such as accuracy, precision, recall, F1-score, and inference time, which have been measured. The comparison of these models will help identify the strengths and weaknesses of each method, both in terms of accuracy in identifying diseases and the speed of the inference process, which is important for field applications. Additionally, precision and recall provide insight into the model's ability to recognize rare or difficult disease classes, while the F1-score combines precision and recall to provide a more comprehensive picture. The results of this comparison are crucial in determining which model is more suitable for rice disease identification and provide a foundation for future system development or improvement. The evaluation comparison between the FK-NN and CNN models is presented in [table 2](#).

**Table 2.** Comparison of FK-NN and CNN Models

Measurement Matrix	FK-NN	CNN Without	FK-NN	CNN
	Without HOG	HOG	With HOG	With HOG
Accuracy	76.00%	85.93%	86.26%	71.37%
Precision	76.25%	85.93%	86.88%	72.74%
Recall	76.00%	85.93%	86.26%	71.37%
F-1 Score	76.05%	85.81%	86.35%	71.20%
Inference Time	0.000013 S	0.000636 S	0.002331 S	0.000282 S

The comparative analysis, showed in [table 2](#), of the FK-NN and CNN models with HOG feature extraction shows that the FK-NN model demonstrates effectiveness in making predictions, while the CNN model shows efficiency in making predictions. Overall, although the CNN model excels in inference time, the FK-NN model performs better in terms of accuracy, precision, recall, and F1-score. The time difference between the two models is not significant, with a difference of only 0.002049 seconds, which does not cause a notable difference in prediction speed. Therefore, the FK-NN model with HOG feature extraction outperforms the CNN model with HOG feature extraction in identifying rice diseases through digital images in this study.

Upon a deeper analysis of the performance metrics, the FK-NN model consistently outperforms the CNN model, especially when HOG feature extraction is applied. The FK-NN model's superior accuracy, precision, recall, and F1-score can be attributed to its sensitivity to the unique features in the rice disease images, which are captured more effectively through the HOG method. HOG's ability to highlight gradients and edge orientations allows the FK-NN model to focus on the crucial textures and patterns on the rice leaves, improving its accuracy in detecting disease categories. Additionally, the CNN model's faster inference time can be attributed to its architecture, which is optimized for speed during prediction. CNNs utilize convolutional layers with shared weights, reducing the number of parameters

compared to fully connected models like FK-NN. This weight-sharing mechanism allows CNNs to process images more quickly, as the model does not need to compute the distance between every test image and all training samples (as is required by FK-NN). Furthermore, CNNs can efficiently exploit hardware acceleration, such as GPUs, enabling faster computations during inference. In contrast, the CNN model, while faster in terms of inference time, seems to be less effective at distinguishing subtle differences in texture and pattern that are essential for accurate classification. The CNN's performance could be hindered by its reliance on learned features that may not be as finely tuned to the specific characteristics of rice diseases, which are critical for achieving high precision and recall. Although CNN shows a slight improvement in speed, the FK-NN model's higher performance in terms of accuracy and other metrics indicates that, for this particular task of rice disease identification, precision and model robustness are more critical than speed.

While both FK-NN and CNN models show promise for rice disease identification, they come with practical challenges. The FK-NN model can become computationally expensive with larger datasets, as it calculates distances between the test sample and every training sample. It's also sensitive to noisy or mislabeled data, which can skew predictions. On the other hand, CNNs, while efficient for large datasets and capable of leveraging hardware acceleration, require significant computational resources for training and may struggle with smaller or less diverse datasets. Both models could face difficulties in real-world scenarios, such as handling variations in lighting, background noise, or image quality. Addressing these challenges will be crucial for improving their effectiveness in practical applications.

#### 4. Conclusion

The study comparing the FK-NN and CNN models for rice disease identification using HOG feature extraction reveals that both models are viable for this task, each with distinct strengths and weaknesses. Performance evaluation was conducted using a confusion matrix, with metrics including accuracy, precision, recall, F1-score, and inference time. The results indicate that the FK-NN model outperforms the CNN model in accuracy, precision, recall, and F1-score. Specifically, the FK-NN model achieved an accuracy of 86.26%, precision of 86.88%, recall of 86.26%, and F1-score of 86.35%, while the CNN model recorded an accuracy of 71.37%, precision of 72.74%, recall of 71.37%, and F1-score of 71.20%. On the other hand, the CNN model demonstrated faster inference times, processing samples in 0.000282 seconds compared to the FK-NN model's 0.002331 seconds. Despite this, the FK-NN model proved more effective in classifying digital images of rice diseases. While the inference time difference of 0.002049 seconds may appear negligible in this study, it is important to consider the trade-offs between speed and accuracy in real-world applications. For instance, in scenarios where rapid decision-making is critical, such as real-time monitoring in agricultural fields, the CNN model's faster inference time could be advantageous despite its lower accuracy. Conversely, in applications where precision and robustness are prioritized, such as detailed disease diagnosis, the FK-NN model's higher accuracy and reliability make it a more suitable choice. Therefore, the selection of a model should depend on the specific requirements of the application, balancing the need for speed against the demand for accuracy. In conclusion, the FK-NN model combined with HOG feature extraction is more effective than the CNN model for identifying rice diseases in digital images, as demonstrated in this study.

To make these models more practical for real-world use in agriculture, there are a few key areas to explore. First, we can improve their ability to handle different lighting, weather, or camera angles by adding more diverse images to the training data or using better preprocessing techniques. Second, tweaking the model settings or trying more advanced designs could help boost accuracy and speed. We could also adapt the models to run on mobile or field devices by making them lighter and faster. Finally, including more disease types and rare cases in the dataset would help the models perform better in a wider range of situations. These steps would make the models more reliable and useful for farmers.

#### 5. Declarations

##### 5.1. Author Contributions

Conceptualization: A.W.O.G., P.V.J.A.G., and K.D.; Methodology: K.D., A.W.O.G.; Software: P.V.J.A.G., and A.W.O.G.; Validation: A.W.O.G., K.D., and P.V.J.A.G.; Formal Analysis: A.W.O.G., K.D., and P.V.J.A.G.; Investigation: A.W.O.G.; Resources: K.D., and P.V.J.A.G.; Data Curation: P.V.J.A.G.; Writing Original Draft

Preparation: A.W.O.G., K.D., and P.V.J.A.G.; Writing Review and Editing: A.W.O.G., and P.V.J.A.G.; Visualization: A.W.O.G. All authors have read and agreed to the published version of the manuscript.

## 5.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

## 5.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

## 5.4. Institutional Review Board Statement

Not applicable.

## 5.5. Informed Consent Statement

Not applicable.

## 5.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

- [1] W. Liu, K. Wang, Y. Zhao, Y. Shen, C. Zhang, Y. Peng, X. Ran, H. Guo, Y. Ding, S. Tang, "Effects of nitrogen application on physicochemical properties of rice starch under elevated temperature," *Food Chem*, vol. 433, no. 02, pp. 1-9, Feb. 2024, doi: 10.1016/j.foodchem.2023.137303.
- [2] T. Sitaresmi, A. Hairmansis, Y. Widyastuti, R. Rachmawati, U. Susanto, B. Wibowo, M. Widiastuti, I. Rumanti, W. Suwarno, Y. Nugraha, "Advances in the development of rice varieties with better nutritional quality in Indonesia," *J Agric Food Res*, vol. 12, no. 06, pp. 1-8, Jun. 2023, doi: 10.1016/j.jafr.2023.100602.
- [3] BPS-Statistics Indonesia, "Harvested area and rice production in Indonesia 2023 (Preliminary figures)." Accessed: Mar. 03, 2025. [Online]. Available: <https://www.bps.go.id/id/pressrelease/2023/10/16/2037/luas-panen-dan-produksi-padi-di-indonesia-2023--angka-sementara-html>.
- [4] D. Prismantoro, S. Akbari, N. Permadi, U. Dey, A. Anhar, M. Miranti, M. Mispan, F. Doni, "The multifaceted roles of Trichoderma in managing rice diseases for enhanced productivity and sustainability," *J Agric Food Res*, vol. 18, no. 12, pp. 1-12, Dec. 2024, doi: 10.1016/j.jafr.2024.101324.
- [5] H. U. Rehman and R. Atiq, "A disease predictive model based on epidemiological factors for the management of bacterial leaf blight of rice," *Brazilian Journal of Biology*, vol. 84, no. 12, pp. 1-10, 2024, doi: 10.1590/1519-6984.259259.
- [6] S. Kaviyarasan, K. Kavitha, N. Indra, P. Meenakshisundram, and K. Thirukumaran, "Isolation, characterization and metabolic profiling of seed endophyte *B. licheniformis* against *Sarocladium oryzae* in rice," *Plant Science Today*, vol. 12, no. sp1, pp. 1-11, Jan. 2025, doi: 10.14719/pst.5948.
- [7] L. Zhou, M. Mubeen, Y. Iftikhar, H. Zheng, Z. Zhang, J. Wen, R. Khan, A. Sajid, M. Solanki, M. Sohail, A. Kumar, E. Massoud, L. Chen, "Rice false smut pathogen: implications for mycotoxin contamination, current status, and future perspectives," *Front Microbiol*, vol. 15, no. 03, pp. 1-12, Mar. 2024, doi: 10.3389/fmicb.2024.1344831.
- [8] R. Singh, P. Krishnan, C. Bharadwaj, S. Sah, and B. Das, "Optimizing chickpea yield prediction under wilt disease through synergistic integration of biophysical and image parameters using machine learning models," *Sci Rep*, vol. 15, no. 1, pp. 1-18, Feb. 2025, doi: 10.1038/s41598-025-87134-0.
- [9] G. Kaur, Rajni, and J. Singh Sivia, "Development of deep and machine learning convolutional networks of variable spatial resolution for automatic detection of leaf blast disease of rice," *Comput Electron Agric*, vol. 224, no. 09, pp. 1-22, Sep. 2024, doi: 10.1016/j.compag.2024.109210.
- [10] Y. Chimate, S. Patil, K. Prathapan, J. Patil, and J. Khot, "Optimized sequential model for superior classification of plant disease," *Sci Rep*, vol. 15, no. 1, pp. 1-12, Jan. 2025, doi: 10.1038/s41598-025-86427-8.

- [11] J. D. Omaye, E. Ogbuju, G. Ataguba, O. Jaiyeoba, J. Aneke, and F. Oladipo, "Cross-comparative review of machine learning for plant disease detection: apple, cassava, cotton and potato plants," *Artificial Intelligence in Agriculture*, vol. 12, no. 6, pp. 127–151, Jun. 2024, doi: 10.1016/j.aiaa.2024.04.002.
- [12] J. M. Cadenas, M. C. Garrido, R. Martínez-España, and M. A. Guillén-Navarro, "Making decisions for frost prediction in agricultural crops in a soft computing framework," *Comput Electron Agric*, vol. 175, no. 8, pp. 1-12, Aug. 2020, doi: 10.1016/j.compag.2020.105587
- [13] Z. Rayan, S. Samir, D. Abdelfattah, and A.-B. M. Salem, "Smart potato disorders diagnostic system based on fuzzy K-nearest neighbor," in *2022 5th International Conference on Computing and Informatics (ICCI)*, IEEE, Mar. 2022, vol. 2022, no. 03, pp. 029–034. doi: 10.1109/ICCI54321.2022.9756104.
- [14] Z. X. Li, X. P. Yan, C. Q. Yuan, and L. Li, "Gear multi-faults diagnosis of a rotating machinery based on independent component analysis and fuzzy K-nearest neighbor," *Adv Mat Res*, vol. 108–111, no. 1, pp. 1033–1038, May 2010, doi: 10.4028/www.scientific.net/AMR.108-111.1033
- [15] X. Chen, H. Zeng, and Z. Li, "A multi-fault diagnosis method of rolling bearing based on wavelet-PCA and fuzzy K-nearest neighbor," *Applied Mechanics and Materials*, vol. 29–32, no. 08, pp. 1602–1607, Aug. 2010, doi: 10.4028/www.scientific.net/AMM.29-32.1602
- [16] J. Padhi, K. Mishra, A. Ratha, S. Behera, P. Sethy, and A. Nanthaamornphong, "Enhancing paddy leaf disease diagnosis -a hybrid CNN model using simulated thermal imaging," *Smart Agricultural Technology*, vol. 10, no. 03, pp. 1-15, Mar. 2025, doi: 10.1016/j.atech.2025.100814
- [17] K. N. Rahman, S. C. Banik, R. Islam, and A. Al Fahim, "A real time monitoring system for accurate plant leaves disease detection using deep learning," *Crop Design*, vol. 4, no. 1, pp. 1-16, Feb. 2025, doi: 10.1016/j.cropl.2024.100092.
- [18] Z. Wang, Y. Wei, C. Mu, Y. Zhang, and X. Qiao, "Rice disease classification using a stacked ensemble of deep convolutional neural networks," *Sustainability*, vol. 17, no. 1, pp. 1-14, Dec. 2024, doi: 10.3390/su17010124
- [19] S. Tan, J. Liu, H. Lu, M. Lan, J. Yu, G. Liao, Y. Wang, Z. Li, L. Qi, X. Ma, "Machine learning approaches for rice seedling growth stages Detection," *Front Plant Sci*, vol. 13, no. 06, pp. 1-15, Jun. 2022, doi: 10.3389/fpls.2022.914771
- [20] H. Nguyen-Quoc and V. Truong Hoang, "Rice seed image classification based on HOG descriptor with missing values imputation," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 18, no. 4, pp. 1897-1903, Aug. 2020, doi: 10.12928/telkomnika.v18i4.14069
- [21] R. Rajakumar, D. Vaishnavi, R. Ramesh, and J. Ganesh, "Optimized Strategy for Rice Plant Disease Detection Using Convolutional Neural Networks," *International Journal of Electronics and Communication Engineering*, vol. 11, no. 7, pp. 208–219, Jul. 2024, doi: 10.14445/23488549/IJECE-V11I7P121.
- [22] J. C. Mary and M. Suganthi, "MRI intracranial neoplasm classification using hybrid LOA-based deep learning classifier," *Biomed Signal Process Control*, vol. 104, no. 06, pp. 1-18, Jun. 2025, doi: 10.1016/j.bspc.2025.107560.
- [23] A. Saini, N. S. Gill, P. Gulia, A. K. Tiwari, P. Maratha, and M. A. Shah, "Smart crop disease monitoring system in IoT using optimization enabled deep residual network," *Sci Rep*, vol. 15, no. 1, pp. 1-21, Jan. 2025, doi: 10.1038/s41598-025-85486-1
- [24] S. Patel and S. Srinath, "Performance Evaluation of Feature Extraction Algorithms for Vehicle Shape Classification," *U.Porto Journal of Engineering*, vol. 8, no. 6, pp. 62–75, 2022, doi: 10.24840/2183-6493\_008.006\_0005.
- [25] W. Jia, S. Chen, L. Yang, G. Liu, C. Li, Z. Cheng, G. Wang, X. Yang, "Ankylosing spondylitis prediction using fuzzy K-nearest neighbor classifier assisted by modified JAYA optimizer," *Comput Biol Med*, vol. 175, no. 06, pp. 1-26, Jun. 2024, doi: 10.1016/j.compbimed.2024.108440
- [26] J. Xing, C. Li, P. Wu, X. Cai, and J. Ouyang, "Optimized fuzzy K-nearest neighbor approach for accurate lung cancer prediction based on radial endobronchial ultrasonography," *Comput Biol Med*, vol. 171, no. 03, pp. 1-17, Mar. 2024, doi: 10.1016/j.compbimed.2024.108038.
- [27] M. Elkharradly, K. Amin, O. M. Abo-Seida, and M. Ibrahim, "Bayesian optimization enhanced FKNN model for Parkinson's diagnosis," *Biomed Signal Process Control*, vol. 100, no. 02, pp. 1-16, Feb. 2025, doi: 10.1016/j.bspc.2024.107142
- [28] S. Sakthipriya and R. Naresh, "Precision agriculture based on convolutional neural network in rice production nutrient management using machine learning genetic algorithm," *Eng Appl Artif Intell*, vol. 130, no. 04, pp. 1-11, Apr. 2024, doi: 10.1016/j.engappai.2023.107682.

- [29] R. R, V. D, R. R, and G. J, "Optimized strategy for rice plant disease detection using convolutional neural networks," *International Journal of Electronics and Communication Engineering*, vol. 11, no. 7, pp. 208–219, Jul. 2024, doi: 10.14445/23488549/IJECE-V11I7P121.
- [30] K. Mahadevan, A. Punitha, and J. Suresh, "A real time performance comparison of rice plant disease identification system using deep CNN models," *International Research Journal of Multidisciplinary Scope*, vol. 05, no. 03, pp. 249–258, Jul. 2024, doi: 10.47857/irjms.2024.v05i03.0159.
- [31] S. Akter, R. I. Sumon, H. Ali, and H.-C. Kim, "Utilizing convolutional neural networks for the effective classification of rice leaf diseases through a deep learning approach," *Electronics (Basel)*, vol. 13, no. 20, pp. 1-19, Oct. 2024, doi: 10.3390/electronics13204095
- [32] Y. Wang, Y. Li, Y. Song, and X. Rong, "The influence of the activation function in a convolution neural network model of facial expression recognition," *Applied Sciences*, vol. 10, no. 5, pp. 1-20, Mar. 2020, doi: 10.3390/app10051897
- [33] K. Goh, S. Surono, M. Afiatin, K. Mahmudah, N. Irsalinda, M. Chaimanee, C. Onn, "Comparison of activation functions in convolutional neural network for poisson noisy image classification," *Emerging Science Journal*, vol. 8, no. 2, pp. 592–602, Apr. 2024, doi: 10.28991/ESJ-2024-08-02-014.