Optimizing Rice Disease Classification Method: A Comparative Study of Artificial Neural Network and Support Vector Machine Using Image Color, Shape, and Texture Features for Improved Efficiency

Joshua R. Dela Cruz
College of Engineering
University of the Philippines
Diliman
jrdelacruz12@up.edu.ph

Vince Raphael R.
Miranda
College of Engineering
University of the Philippines
Diliman
vrmiranda@up.edu.ph

Prospero C. Naval, Jr.
Department of Computer
Science
University of the Philippines
Diliman
pcnaval@up.edu.ph

ABSTRACT

Rice plays a pivotal role in the Philippines, serving as a staple food and a significant contributor to the country's agricultural and economic landscape. Despite its importance, the rice industry faces multiple challenges, with rice disease as one of the leading causes of reductions in rice yield. The current studies on rice disease classification using ANN and SVM often uses rice disease image pre-processing and segmentation procedures to remove noise and isolate the disease portions of a rice plant. From the disease segmented portion, features are extracted which are then used for training and classifying rice diseases. This method may tend to be problematic as it is inefficient and often requires human intervention to optimize different sets of segmentation parameters before classifying images. This study introduces an automatic and more efficient method in classifying rice disease images by utilizing the whole image and removing the image preprocessing and segmentation step altogether. To classify rice disease images, the researchers made use of color, shape, and texture features. These features were used to train both SVM and ANN models, the top performing models in the literature, for comparison. The models are then evaluated using accuracy as classification performance metrics.

Keywords

computer vision, image classification, rice plant diseases, artificial neural network, support vector machine

1. INTRODUCTION

Rice, often referred to as the "staple of life" in the Philippines, holds unparalleled significance deeply rooted in the country's culture and sustenance [3]. The reliance on rice in the Philippines is immense, with the nation's daily diet revolving around its consumption. In 2023 alone, the domestic consumption for milled rice reached a staggering 16.5 million metric tons [14]. Filipino farmers rely on rice cultivation to provide not only for their families but for the entire nation. The importance of rice is evident, as it remains the primary choice for staple food in the Philippines despite the availability of alternative options like noodles and bread [18]. While highly significant, rice's global production continuously faces substantial challenges due to the impact of rice diseases. A 2016 study conducted in the United States reveals that approximately 30% of rice crop production losses worldwide can be attributed to diseases, creating a significant deficit equivalent to the sustenance of about 60 million people [15]. Diseases such as bacterial blight, can inflict up to a staggering 70% yield loss, with the severity increasing especially if it strikes early in the crop's life cycle [2].

To address the challenge of yield loss, rice disease management strategies are crucially employed, involving intricate processes of detection, classification, and treatment. Early classification and forecasting systems, essential for managing rice diseases like rice blast, guide farmers in strategic decision-making for fungicide applications, fertilization practices, and yield prediction. Robust early-warning systems can prevent explosive disease outbreaks, reducing both yield losses and environmental impacts [10].

The primary hurdle lies in accurately classifying rice diseases, especially in vast fields with non-native pathogens thus prompting a shift towards the adoption of image processing for precise identification [21, 9]. Application of machine-driven systems can also reduce expert and labor costs by lowering the need for farmers to manually check the health condition of their crops. All in all, this will contribute to the sustainable agriculture efforts in the Philippines and ultimately ensure a stable food supply for the nation.

Section 2 of this paper provides multiple related existing studies that leverage computer vision for rice disease classification. Previous research in this domain has typically focused on classifying a limited number of diseases, usually ranging from one to three per classification model. Common procedures involve multiple pre-processing steps, such as background removal or isolating the diseased portion of the rice plant before extracting features. In contrast, this paper introduces a classification method that eliminates the need for pre-processing, capitalizing on multiple global features of an image. Furthermore, the study extends the scope of classification to encompass a broader range of common Philippine rice diseases.

2. RELATED STUDIES

Machine learning techniques for rice disease identification have been utilized since the late 2000s. In a comprehensive survey by Shah, J. P. et al. [19], 19 papers were analyzed, providing insights into the landscape of research on rice plant diseases. The survey compared various criteria, including the size of the image dataset, the number of classes or diseases considered, preprocessing and segmentation techniques, and the input features used. Different classifiers such as Nearest Neighbors, Support Vector Machines, and Neural Networks were employed, achieving accuracies ranging from 75% to 100%.

In a Philippines-based study, the research focused on three prevalent diseases—Bacterial leaf blight, Brown spot, and Rice blast. The methodology employed MATLAB functions, implementing a two-step image enhancement process for noise removal and a five-step segmentation technique to isolate the diseased portion before feature extraction. The chosen features encompassed color moments in both RGB and HSV color spaces while the classification model used was a traditional Neural Network with Backpropagation. Utilizing a database of 134 disease images of leaf samples acquired using a controlled-light module box which was designed with specific dimensions to capture the required length of rice leaf, ensured controlled lighting for image acquisition, the model successfully identified typical lesions, achieving 100% accuracy in disease identification [17].

3. METHODOLOGY

This section outlines the methodologies employed in conducting the study. As the research emphasizes minimal preprocessing steps, the discussion will delve into various image features then focus on the ones utilized in the study. The experimental setup will be presented to provide insights into the parameters and tools used. Finally, a comprehensive detail of the study's implementation will be discussed.

3.1 Dataset

The dataset used in this study was retrieved from Kaggle[1], a subsidiary of Google and an online community of data scientists and machine learning engineers which allows users to share and find datasets they want to use in building AI models, publish datasets, and work with other data scientists and machine learning engineers. The obtained dataset[2], originally from Omdena's Local Chapter project - Creating a Rice Disease Classifier using Open Source Data and Computer Vision, comprises two folders, namely 'extra_resized_raw_images' and 'resized_raw_images,' each containing diverse images of rice plants (paddy images, zoomed-in images, processed images). All images are standardized to a dimensionality of 224 × 224 pixels. Within the dataset, there are 13 distinct rice diseases categorized into three groups:

1. Fungal: Affects the leaf blade, sheath, stem, node, and panicle.
2. Bacterial: Affects the leaf blade.
3. Viral: Affects the leaf blade and sheath.

The `'resized_raw_images'` folder, which consists of original images for all 14 classes, including 13 diseases and 1 healthy class, was utilized. This subset has undergone cleaning and evaluation by other users, ensuring the removal of near-duplicate images from different classes. The dataset, as summarized in Table 1, is balanced across the classes, providing a comprehensive representation of the different rice plant conditions.

Table 1: Image Data and Corresponding Class Number

Disease	Class Number	Number of Images
bacterial leaf blight	0	97
bacterial leaf streak	1	99
bakanae	2	100
brown spot	3	100
grassy stunt virus	4	100
healthy rice plant	5	100
narrow brown spot	6	98
ragged stunt virus	7	100
rice blast	8	98
rice false smut	9	99
sheath blight	10	98
sheath rot	11	91
stem rot	12	100
tungro virus	13	100

For the experiment proper, three different datasets were created from the original data.

1. First Dataset - Main dataset partitioned into training and testing sets. 80% of the dataset, comprising 1024 diseased and 80 non-diseased rice images, was allocated for training. The remaining 20%, consisting of 256 diseased and 20 non-diseased rice images, was designated for testing.
2. Second Dataset - Contains only three classes: brown spot, bacterial leaf blight, and rice blast; created for the purpose of comparing the performance of this study's ANN and SVM results relative to the previous local study conducted. To create a similar setting to the local study as will be discussed in Section 3.3, the same amount of images focusing on the rice plant were randomly selected for each class, removing paddy images in the dataset resulting in 50 images per class. The total images were then partitioned into an 80-20 ratio for training-testing, similar to the First Dataset.
3. Third Dataset - Similar to the first dataset, 14 classes will be classified but in order to mimic a similar setting to most papers that only used similar types of images

[1] https://www.kaggle.com
[2] https://www.kaggle.com/datasets/shrupyag001/philippines-rice-diseases

per class, 50 randomly selected images will be selected for each class, focusing on only either zoomed-in or paddy shot images per class.

3.2 Image Feature

An image feature, a value derived from a predefined image-processing algorithm, characterizes the content of an image. Global features encompass information from the entire image, while local features focus on specific regions corresponding to objects within the image. Image content is thus the aggregation of all possible features or combinations of basic features over a target image

Mathematically, a feature is an n-dimensional vector, computed through image analysis, with commonly used cues such as color, texture, shape, and spatial information. Aiming to develop a simple model algorithm independent of image preprocessing and adaptable to various image types, we opt for global features of color, texture, and image moments.

3.2.1 Color Features

Color stands out as the most expressive among visual features, garnering extensive attention in image-related studies over the past decade. Crucial in computer vision, color ranks among the most significant and widely utilized low-level features. Color features offer a more domain- independent nature due to their minimal semantic meaning and compact representation. The initial step in extracting color features involves selecting an appropriate color space, with options including RGB (Red, Green and Blue), CMYK (Cyan, Magenta, Yellow, and Black), HSV (Hue, Saturation, and Value), CIE Luv* (CIE Lightness, U*, and V*), and , and CIE Lab* (CIE Lightness, a* and b*) [1].

In the realm of computer vision, numerous color descriptors are available for extracting the color features of an image. Table 2 offers an overview of various color methods, as sourced from the literature [22], outlining their respective strengths and weaknesses. Aligning with our objective of devising a simple algorithm with minimal computational complexity, our analysis focuses on color histogram and color moments of RGB, HSV and CIE Lab*.

Color histogram is a type of line graph, where each peak represents a particular color of the color space being used. The peaks in a color histogram are referred to as bins and they represent the x-axis. The number of bins depends on the number of colors there are in an image. The y-axis denotes the number of pixels there are in each bin. In other words it denotes how many pixels in an image are of a particular color [16].

Color moments offer computational simplicity and minimal storage. The mathematical meaning of this approach is that any color distribution can be interpreted as a probability distribution [11]. In this paper, the moments such as the mean (first order) (1), standard deviation (second order) (2), skewness (third order) (3), and kurtosis (fourth order) (4) represent the feature vector and defined as follows:

Table 2: Comparison of Different Color Descriptors

Color Method	Pros	Cons
Histogram	Simple to compute, intuitive	High dimension, no spatial info, sensitive to noise
Color Moment	Compact, robust	Not enough to describe all colors, no spatial info
Color Coherence Vector (CCV)	Spatial info	High dimension, high computation cost
Correlogram	Spatial info	Very high computation cost, sensitive to noise, rotation and scale
Dominant Color Descriptor (DCD)	Compact, robust, perceptual meaning	Post-processing is needed for spatial info
Color Structure Descriptor (CSD)	Spatial info	Sensitive to noise, rotation and scale
Scalable Color Descriptor (SCD)	Compact on need, scalability	No spatial info, less accurate if compact

1. Mean:
$$\mu_i = \frac{1}{N}\sum_{j=1}^{N} e_{ij} \tag{1}$$

2. Standard Deviation:
$$\sigma_i = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(e_{ij}-\mu_i)^2} \tag{2}$$

3. Skewness:
$$\gamma_i = \frac{\frac{1}{N}\sum_{j=1}^{N}(e_{ij}-\mu_i)^3}{\left(\sqrt{\frac{1}{N}\sum_{j=1}^{N}(e_{ij}-\mu_i)^2}\right)^3} \tag{3}$$

4. Kurtosis:
$$\kappa_i = \frac{\frac{1}{N}\sum_{j=1}^{N}(e_{ij}-\mu_i)^4}{\left(\sqrt{\frac{1}{N}\sum_{j=1}^{N}(e_{ij}-\mu_i)^2}\right)^4} \tag{4}$$

The color value of the i-th color component of the j-th image pixel is represented as e_{ij}, and N signifies the total number of pixels in the image.

3.2.2 Texture Features

For this study, textural features were derived from the Grey Level Co-occurrence Matrix (GLCM), commonly known as Haralick features [5]. GLCM incorporates an offset value (D) to define pixel adjacency based on a specific distance, and in this case, the offset value is set to 1. The features, computed using the GLCM matrix, are summarized in Table 3 [4].

Table 3: Mahotas Haralick Features

Feature Name	**Formula**	**Definition**
'Angular Second Moment'	ASM $= \sum_i \sum_j P(i,j)^2$	Represents the uniformity of distribution of grey level in the image
'Contrast'	Con $= \sum_i \sum_j \lvert i-j \rvert^2 P(i,j)$	Local variations to show the texture fineness
'Correlation'	Corr $= \frac{\sum_i \sum_j (i-\mu)(j-\nu)P(i,j)}{\sigma(i)\sigma(j)}$	Linear dependence in GLCM between different index
'Sum of Squares: Variance'	SS: Var $= \sum_i \sum_j (i-\mu)^2 P(i,j)$	Higher weights that differ from the average value of GLCM
'Inverse Difference Moment'	IDM $= \sum_i \sum_j \frac{1}{1+\lvert i-j \rvert} P(i,j)$	Inverse Contrast Normalized
'Sum Average'	(Mean sum $= \sum_i \sum_{k=2}^{2N} kP_{x+y}(k)$	Higher weights to the higher index of the marginal GLCM
'Sum Variance'	Var sum $= \sum_i \sum_{k=2}^{2N} (k-\mu_{x+y})^2 P_{x+y}(k)$	Higher weights that differ from the entropy value of the marginal GLCM
'Sum Entropy'	H sum $= -\sum_i \sum_{k=2}^{2N} P_{x+y}(k) \log(P_{x+y}(k)+\epsilon)$	Higher weight on the higher sum of the index entropy value
'Entropy'	H $= -\sum_i \sum_j P(i,j) \log(P(i,j)+\epsilon)$	Texture randomness producing a low value for an irregular GLCM
'Difference Variance'	Var diff $= \sum_i \sum_{k=0}^{N-1} k^2 P_{x-y}(k)$	Average of the squared differences from the mean
'Difference Entropy'	H diff $= -\sum_i \sum_{k=0}^{N-1} P_{x-y}(k) \log(P_{x-y}(k)+\epsilon)$	Higher weight on the higher difference of the index entropy value
'Informational Measure of Correlation 1'	IMC1 $= \frac{HXY1-HXY2}{\max(HX,HY)}$	Entropy measures
'Informational Measure of Correlation 2'	IMC2 $= \sqrt{1-\exp(-2(HXY2-HXY1))}$	Entropy measures

i and j represent rows and columns respectively, N is the number of distinct grey levels in the quantized image, $P(i,j)$ is the element from the normalized GLCM matrix, $P_x(i)$ and $P_y(j)$ are marginal probabilities of the matrix obtained by summing rows and columns of GLCM, respectively.

3.2.3 Shape Features

For this study, shape was obtained through image moments. Image moments serve as comprehensive descriptors capturing the global features of an image, offering insights into various geometrical characteristics. In computer vision, moments prove effective for characterizing diverse geometric attributes of an object, including its area, border, location, and orientation. The utilization of moments in image analysis and pattern recognition found inspiration in the work of Hu [7]. Over the years, the field of computer vision has delved into multiple moment descriptors, Moment Invariants, Geometric Moments, Rotational Moments, Orthogonal Moments, and Complex Moments [13].

In this study, the two most common orthogonal moments, Zernike Moment and Legendre Moment are utilized due to their desirable property of orthogonality, which ensures their effectiveness in describing images with mutually independent descriptors, minimizing information redundancy. They are also invariant to translations and rotation.

Zernike Moment is the mapping of an image onto a set of complex Zernike polynomials. This is basically seen as an inner product between the image's function and the Zernike polynomials [**?**].

The two-dimensional Zernike moment of an image intensity function $f(r, \theta)$ is defined as:

$$Z_{pq} = \frac{p+1}{\pi} \int_0^{2\pi} \int_1^{-} V_{pq}(r,\theta)^* f(r,\theta) r \, dr \, d\theta \tag{5}$$

where

$$V_{pq}(r,\theta) = R_{pq}(r) e^{jq\theta} \tag{6}$$

and

$$R_{pq}(r) = \sum_{s=0}^{\frac{p-q}{2}} (-1)^s \frac{(p-s)!}{s! \left(\frac{p-2s+q}{2}\right)! \left(\frac{p-2s-q}{2}\right)!} r^{p-2s} \tag{7}$$

with $0_- < |p| < q$ and $p - |q|$ is given; $p > 0$.

Legendre Moment is first presented by Teague [20] which introduces the Legendre polynomials that form a complete orthogonal set inside the unit circle. The kernel of the Legendre moments are therefore defined as products of Legendre polynomials defined along rectangular image coordinate axes inside a unit circle.

The two-dimensional Legendre moment of an image intensity function f(x,y) is defined as:

$$\mathrm{L}_{pq} = \frac{(2p+1)(2q+1)}{4} \int_{-1}^{1} \int_{-1}^{1} \mathrm{p}_p(x) \mathrm{p}_p(y) f(x,y) \, dx \, dy \tag{8}$$

The kernel functions P, denote Legendre polynomials of order n:

$$\mathrm{p}_p(x) = \frac{1}{2^p} \sum_{k=0}^{\frac{p}{2}} (-1)^k \frac{(2p-2k)!}{(k)!(p-k)!(p-2k)!} x^{p-2k} \tag{9}$$

3.3 Experimental Setup

3.3.1 Experiment

To assess the effectiveness of global features obtained without preprocessing the images, two classification models: the Artificial Neural Network (ANN) and Support Vector Machine (SVM) will be employed. The performance of both models will then be evaluated using the datasets presented in Section 3.1 under the following conditions:

1. Generalizability of the models across 14 classes (13 diseased and 1 non-diseased rice images) and diverse image types (paddy images, zoomed-in images, processed images) using the first dataset. This will also investigate whether the selected attributes are sufficient for ANN and SVM, considering the dataset's diversity.
2. Using the second dataset, a comparative evaluation with the local literature [20] which achieved 100% accuracy using ANN for three rice diseases (brown spot, rice blast, and leaf blight) which serves as the benchmark for the modes of this study.
3. Test under the third dataset to compare the performance of the ANN and SVM models to common dataset settings used by international literatures wherein there is less variability in the image magnification for each class i.e. each class is solely composed of either zoomed in or whole rice plant images.

3.3.2 Hyperparameters

Artificial Neural Network. ANN requires multiple hyperparameters to tune to create a good result based on the input features. An initial configuration for the network architecture and hyperparameters were set arbitrarily as presented in Table 4. This was used to initially grasp the potential of the extracted features, and to evaluate them during the selection process in the Principal Component Analysis.

Table 4: Initial Hyperparameter Values for ANN

Hyperparameter	**Value**
Number of Hidden Layers	2
Nodes per Hidden Layer	32-32
Activation Functions	relu-relu
Learning Rate	0.001
Batch Size	32

Subsequently, the parameters were then tuned using keras-tuner's hyperband[3] [12] under the identified set of values presented in Table 5.

Table 5: Hyperparameter to tune for ANN

Hyperparameter	**Values**	**Step**
Number of Hidden Layers	1 to 3	1
Nodes per Hidden Layer	32 to 512	32
Activation Functions	relu, tanh, sigm	N/A
Learning Rate	0.001, 0.01, 0.1, 0.3, 0.5	N/A
Batch Size	16, 32, 64, 128	N/A

Support Vector Machine. Building a support vector machine also requires its own set of parameters, namely: kernel, C,

[3] https://keras.io/api/keras_tuner/tuners/hyperband/

and gamma. This set of hyperparameters affects the mapping, accuracy, and generalization of the model. For the initial model the researchers make use of the preset parameters when building a support vector machine using scikit-learn[4]. These initial values are referenced in Table 6. This provides researchers with a baseline idea on the performance of the model before any feature selection or hyperparameter tuning is done.

Table 6: Initial Hyperparameter Values for SVM

Hyperparameter	Value
Kernel	Linear
C	1
Gamma	scale

After selecting the optimal features, the set of parameters presented in Table 7 are then tuned using gridsearch[5]

Table 7: Hyperparameters to Tune for SVM

Hyperparameter	Values
Kernels	rbf, linear, poly
C	0.001, 0.01, 0.1, 1
Gamma	1, 10, 100, 300

3.3.3 Implementation

The general high-level overview of the implementation for each experiment that was performed are shown in Figure 1. The specific details for each experiment are discussed in the subsequent sections.

Data Preparation

In implementing the project, the researchers first prepared the data as detailed in Section 3.1. The jpeg images were then translated to BGR images before converting to RGB, which is often the input data used to extract the aforementioned image moments, color, and texture features. The conversion was done by making use of OpenCV's CV2[6].

Feature Extraction

Texture. Texture extraction was achieved by first converting the RGB image data into grayscale [source description] with CV2. The researchers then utilized the Mahotas Feature[7] library to get the following haralick features from each image which were all appended to form a GLCM feature dataframe.

Histogram. Histogram was calculated by converting the RGB images to HSV and LAB color spaces using CV2. Once converted, the researchers then proceed to split 3-dimensional tensor color spaces by their three respective channels. For each channel the minimum and maximum values were taken and the number of bins parameters was set. The regions, which is the difference of the minimum and maximum values divided by the number of bins, then became the features with their values being the counts of numbers in each channel that fit within the range. As the 'number of bins' parameter affects the number of histogram features created with more bins leading to more histogram color features for each channel, the researchers experimented with different values ranging from zero to one hundred and tested the accuracy of the SVM model for each number of bins to determine the the optimal and final number of bins to be used in the model

Color Moments. Color moments were also computed using the RGB, HSV, and LAB channels. From each channel of the aforementioned color spaces, the researchers computed the mean, variance, kurtosis, and skewness of each image using the numpy libraries for mean and variance while using scipy[8] for skewness and kurtosis.

Image Moments. Mahotas Feature was also used to extract the Zernike Moments while scipy's special[9] was employed to obtain the Legendre Moments. Traditionally, various degrees of order are evaluated to determine the optimal value for each, but multiple literatures suggest that the performance of a model plateau after reaching a certain point of order for the Zernike and Legendre Moments [6, **?**], especially if there are minimal smooth color changes [8]. Additionally, computing for higher order moments is computationally expensive and does not align with the objective of this research [16]. Hence for this study, a fixed value was chosen as referred to the literature for these features.

One Hot Encoding and Train Test Split

All the feature dataframes along with their labels are then concatenated into a single dataframe. Where the features are normalized using sklearn's minmax scaler[10]. The class names are then converted into integers for model training using sklearn's label encoder[11]. This will create the final input dataframe. The final input data is then split into train test splits using sklearns train_test_split[12] function.

Artificial Neural Network

For the implementation of the ANN, Keras from Tensorflow[13] was used. Preliminary architecture settings and hyperparameters as mentioned in Section 3.3.2 were set arbitrarily to create a simple model. This was then trained using the training data that was split into 95%-5% training-validation sets and evaluated using the test data with complete features. Subsequently, principal component analysis (PCA) was used to select a subset of best features from the initial dataframe which were used for the second evaluation of the model. Finally, the model with the tuned hyperparameters was tested after training on the training data with refined features.

[4] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

[5] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

[6] https://docs.opencv.org/3.4/d8/d01/group__imgproc__color__conversions.html

[7] https://mahotas.readthedocs.io/en/latest/features.html

[8] https://docs.scipy.org/doc//reference/stats.html

[9] https://docs.scipy.org/doc/scipy/reference/generated/scipy.special.legendre.html

[10] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

[11] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html

[12] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

[13] https://www.tensorflow.org/guide/keras

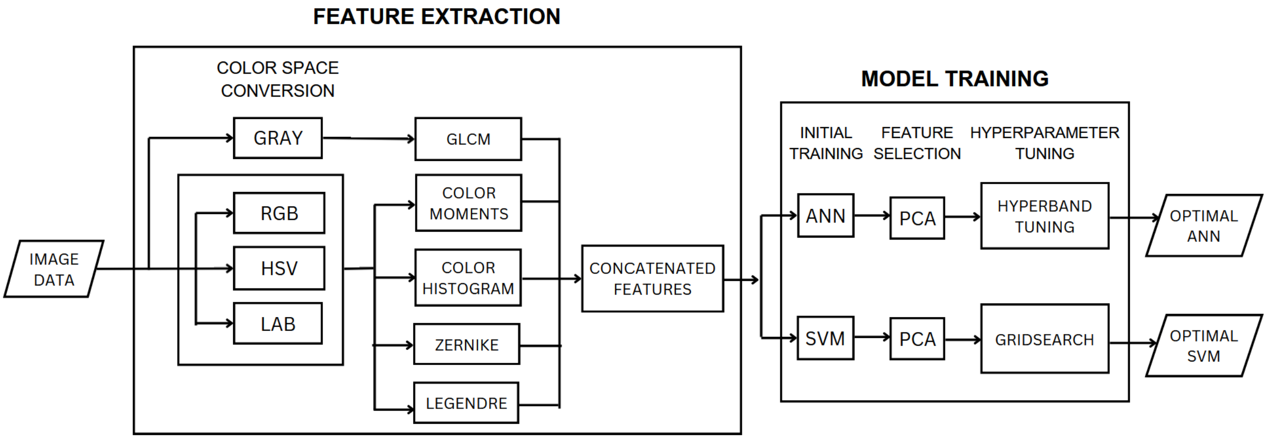


Figure 1: Experiment Implementation

Support Vector Machine
An initial Support Vector Machine (SVM) model is built using the sklearn library. The model is trained using the train data with complete features and the accuracy is computed with using test data. Once the initial model has been trained, the researchers then proceed to feature selection making use of principal component analysis (PCA) experimenting with a range of variances and number of features [source description]. These sets of features are then used to build multiple models which are trained and evaluated using the parameters of the initial SVM model and the model features with the best test accuracy are then used for hyperparameter tuning. Hyperparameter tuning is then done with gridsearch. Gridsearch is a machine learning algorithm that tests the performance of all possible combinations of listed hyperparameters for researchers to find the optimal hyperparameter setting to use in their model [source]. The best model is then selected after evaluating each model by their accuracy, precision, recall, and F-1 score [source].

4. DISCUSSION AND ANALYSIS OF RESULT

4.1 Experiment 1: Assessing Generalizability Across Diverse Dataset and Multiple Classes

The initial experiment discussed in Section 3.3.1 was executed and a total of 933 features were obtained as broken down in Table 8

Table 8: Feature Descriptors and Dimensions

Descriptor	Color Channel	Dimension per Channel	Total Features
Histogram	RGB	82	246
Histogram	HSV	82	246
Histogram	LAB	82	246
Color Moment	RGB	4	12
Color Moment	HSV	4	12
Color Moment	LAB	4	12
GLCM	Gray	13	13
Zernike	Gray	25	25
Legendre	Value	121	121

To categorize the images into diseased and non-diseased classes, the ANN and SVM classifiers were used. As an initial assessment, the performance of both models was evaluated using individual sets of features. Each run focused on either histogram, color moment, texture, Legendre moments, or Zernike moments only, with the result presented in the Table 9. Accuracies are based on models with selected best features for each set and tuned parameters.

Table 9: Performance of ANN and SVM using Individual Sets of Features

Feature	ANN (%)	SVM (%)
Histogram	78.62	85.14
Color Moment	76.09	55.79
GLCM	60.87	59.78
Zernike	13.04	13.85
Legendre	38.41	41.3

4.1.1 Artificial Neural Network

The result of the three evaluations under ANN is presented in Table 10, showing that with the complete set of extracted features and unrefined hyperparameters initially mentioned in Section 3.3.2, the model could already achieve a good accuracy of 80.79%. After using PCA to reduce the dimensionality of the data, minimal changes were obtained on the metric. Finally, tuning the hyperparameters with hyperband creates 186 trial combinations wherein the top 10 best results for tuned parameters are shown in Table 11, a small increase on the accuracy resulting in 83.33% was obtained. This implies that either the features obtained for all the 14 classes or that the feature selection performed were not sufficient.

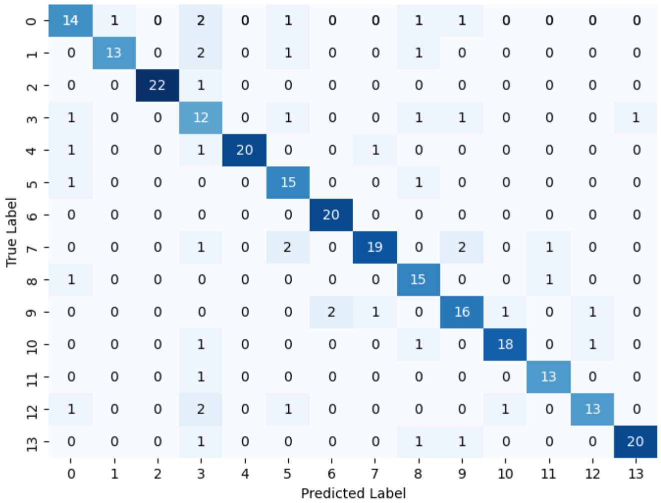
Table 10: Model Comparison for ANN

Model	Features	Hyperparameter	Accuracy (%)
ANN 1	933	untuned	80.79
ANN 2	246	untuned	80.72
ANN 3	246	tuned	83.33

Table 11: Top 10 Trials for the ANN Hyperparameter Tuning

Trial Number	Number of Layers	Nodes per Hidden Layer	Activation Function	Learning Rate	Batch Size	Accuracy (%)
0070	3	480-320-480	relu-tanh-relu	0.1	32	83.93
0076	3	480-320-480	relu-tanh-relu	0.1	32	83.93
0080	3	192-32-512	sigm-tanh-relu	0.1	16	83.93
0106	2	480-352-224	relu-tanh-tanh	0.01	16	83.93
0119	3	64-512-64	tanh-tanh-tanh	0.1	128	83.93
0165	3	384-384-320	sigm-relu-sigm	0.3	16	83.93
0172	3	192-96-512	tanh-sigm-relu	0.5	128	83.93
0066	3	512-128-320	sigm-sigm-tanh	0.1	128	82.14
0067	3	512-128-320	sigm-sigm-tanh	0.1	128	82.14
0073	3	512-128-320	sigm-sigm-tanh	0.1	128	82.14

Additionally, a confusion matrix presented in Figure 2 was created identifying class 6 (narrow brown spot), class 2 (bakanae), and class 11 (sheat roth) as the top 3 classes/diseases that had minimal error during the classification.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	14	1	0	2	0	1	0	0	1	1	0	0	0	0
1	0	13	0	2	0	1	0	0	1	0	0	0	0	0
2	0	0	22	1	0	0	0	0	0	0	0	0	0	0
3	1	0	0	12	0	1	0	0	1	1	0	0	0	1
4	1	0	0	1	20	0	0	1	0	0	0	0	0	0
5	1	0	0	0	0	15	0	0	1	0	0	0	0	0
6	0	0	0	0	0	0	20	0	0	0	0	0	0	0
7	0	0	0	1	0	2	0	19	0	2	0	1	0	0
8	1	0	0	0	0	0	0	0	15	0	0	1	0	0
9	0	0	0	0	0	0	2	1	0	16	1	0	1	0
10	0	0	0	1	0	0	0	0	1	0	18	0	1	0
11	0	0	0	1	0	0	0	0	0	0	0	13	0	0
12	1	0	0	2	0	1	0	0	0	0	1	0	13	0
13	0	0	0	1	0	0	0	0	1	1	0	0	0	20

Figure 2: Confusion Matrix of Best ANN Model

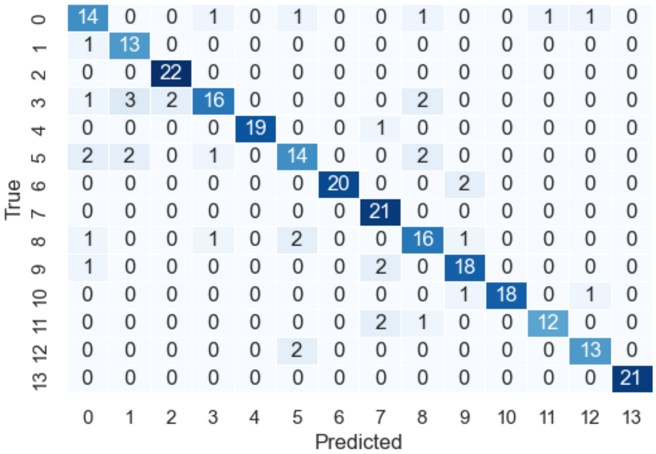
4.1.2 Support Vector Machine

The findings of training the SVM on the three datasets are shown in Table 12. In the initial training the study found an accuracy of 78.62%. Similar to the ANN, even with the absence of hyperparameter tuning and feature selection the model is already adequate at classifying rice diseases. This accuracy, however, was further improved after feature selection via PCA where the SVM reduced the features to 194 from 933 while still achieving a higher accuracy of 84.05%. The highest accuracy was then achieved via hyperparameter tuning making use of gridsearch with the top results presented in Table 13 which tested the outcomes of different kernels, c, and gamma values. The final accuracy of the model was 86.23%.

A confusion matrix shown in Figure **??** was also outputed where the researchers will see that bakanae (2), narrow brown spot (6), ragged stunt virus (7), and tungro virus (13) have the highest class accuracies

Table 12: Model Comparison for SVM

Model	Features	Hyperparameter	Accuracy (%)
SVM 1	933	untuned	78.62
SVM 2	194	untuned	84.05
SVM 3	194	tuned	86.23



	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	14	0	0	1	0	1	0	0	1	0	0	1	1	0
1	1	13	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	22	0	0	0	0	0	0	0	0	0	0	0
3	1	3	2	16	0	0	0	0	2	0	0	0	0	0
4	0	0	0	0	19	0	0	1	0	0	0	0	0	0
5	2	2	0	1	0	14	0	0	2	0	0	0	0	0
6	0	0	0	0	0	0	20	0	0	2	0	0	0	0
7	0	0	0	0	0	0	0	21	0	0	0	0	0	0
8	1	0	0	1	0	2	0	0	16	1	0	0	0	0
9	1	0	0	0	0	0	0	2	0	18	0	0	0	0
10	0	0	0	0	0	0	0	0	0	1	18	0	1	0
11	0	0	0	0	0	0	0	2	1	0	0	12	0	0
12	0	0	0	0	0	2	0	0	0	0	0	0	13	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	21

Figure 3: Confusion Matrix of Best SVM Model

4.2 Experiment 2: Comparative Evaluation with the Local Study

In the second experiment, using the dataset comprising randomly selected zoomed-in images, a new dataframe of features was extracted, yielding an initial set of 933 features. Similar to Experiment 1, the initial ANN and SVM models were employed to evaluate the features through PCA. Subsequently, each model was tuned to obtain optimal parameters then utilizing these models, the accuracy results for individual rice diseases and overall performance are presented in Table 14 below.

It can be observed that for both local literature and this study's proposed method, a 100% accuracy was obtained when classifying the three classes of rice disease. On the other hand, SVM only attained a 90.91% accuracy when classifying rice blast images.

Table 13: Top 10 Trials for the SVM Hyperparameter Tuning

Trial Number	C	Gamma	Kernel	Accuracy (%)
0	100	0.1	rbf	0.862319
25	100	0.1	rbf	0.862299
16	10	0.1	rbf	0.860481
22	100	0.01	rbf	0.842361
11	10	0.001	linear	0.822456
17	10	0.1	linear	0.822456
14	10	0.01	linear	0.822456
27	100	0.1	poly	0.821526
5	1	0.01	linear	0.820633
2	1	0.001	linear	0.820633

Table 14: Comparison of Accuracy (in %) Results for ANN of Local Literature (LL) vs ANN and SVM of Proposed Method (PM)

Disease	ANN (LL)	ANN (PM)	SVM (PM)
Brown Spot	100	100	100
Rice Blast	100	100	90.91
Leaf Blight	100	100	100
Overall	100	100	96.77

4.3 Experiment 3: Comparative Evaluation with International Studies

For this experiment, performance for both ANN and SVM revealed very high results as presented in Table 15, as compared to the models of experiment 1 which utilized the dataset with varied types of images for each class, This highlights the effectiveness of the model when classifying single variation of images for each label.

Table 15: Performance of ANN and SVM with Different Models for Experiment 3

Model	ANN (%)	SVM (%)
1	90.07	88.65
2	91.49	93.61
3	93.62	93.61

Model 1: Complete features, untuned parameters
Model 2: Selected features, untuned parameters
Model 3: Selected features, tuned parameters

As can be seen, ANN with all the extracted features and untuned parameters, already obtained a 90.07% accuracy, subsequently performing evaluation on selected features resulted in slight increase resulting to 91.49%. Finally tuning the parameters, the hypermodel obtained 93.62%.

On the other hand, SVM were also able to obtain a very high accuracy of 88.65% immediately even with all the features and untuned parameters. It was increased to 93.61% after selecting the best features and more, reaching a great value of 93.63% after performing gridsearch for its parameters.

5. CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In this study, the researchers sought to develop a machine-driven classification method that eliminates the necessity for pre-processing diverse rice images. The focus was on leveraging multiple global features of an image, specifically Haralick Texture Features, Color Histogram, Color Moments, and the Orthogonal Image Moments, Legendre and Zernike Moments. While numerous studies have addressed rice disease classification, efforts were made to broaden the scope by classifying prevalent rice diseases in the Philippines. The classification included a total of 13 classes representing rice diseases affecting various parts of a rice plant, in addition to images of healthy rice plants.

To assess the explanatory power of the features, an initial checking for individual sets of each feature was conducted showing the effectivity of the color features.

ANN and SVM models were then employed to classify the full set of 933 features and comparing the results of this with just using the features selected by PCA, resulted in minimal difference for ANN but a substantial change for SVM increasing the initial accuracy of 78.62 to 84.05%. Ultimately, using the subset containing the best features and tuned models for both ANN and SVM, an accuracy of 83.33% for ANN and 86.23% for SVM were obtained confirming the effectiveness of SVM in classifying high-dimensional datasets.

Though examining the result of Experiment 2 outlined in Section 4.2, it is evident that the proposed methodology in this study demonstrates success particularly when dealing with a limited number of classes and less variability in images, comparable to the local study that utilized multiple steps to remove noise and isolate the disease portion of rice plant.

Finally, looking at Experiment 3 which was performed as a supplementary assessment. To align with other international rice disease image classification studies that often utilize preprocessed and segmented images, a test was conducted on randomly selected and balanced of either zoomed-in or whole rice field images for each of 14 classes revealing a final result of 93.62% test accuracy for ANN and 93.61% for SVM, highlighting the great potential of the selected features and emphasizing the impact of diverse image types on classification accuracy.

In summary, the proposed method nonetheless, underscores its significant potential in classifying a diverse dataset of rice disease images, including those that include paddy shots which are actually more common in the real world. Remarkably, this classification is achieved with no pre-processing, eliminating the need for intricate steps and complex computations usually associated with common approaches.

5.2 Recommendations

As good as the results obtained in this study are, the researchers believe that are still further areas that could be explored to improve the study:

1. As seen in the results of experiment 1, despite conducting feature selection and optimizing the models with tuned hyperparameters, achieving an accuracy of over 90% was not possible suggesting that the extracted features may be insufficient or that a more effective feature selection method is essential, especially when dealing with a complex dataset with numerous classes. Further exploration of other global features and feature selection techniques are recommended to enhance the model's performance and potentially achieve perfect accuracy in classifying diverse sets of rice disease images.
2. Given that the study performed better in classifying a dataset of only zoomed in rice diseases images, future studies may want to explore the models performance on a dataset composed solely of whole rice plant or paddy images.
3. Furthermore, while less efficient, prior studies employing rice disease image segmentation before feature extraction and classification have demonstrated promising outcomes. Hence, prospective researchers may want to explore the extraction of features used in this study on pre-segmented rice images.

6. REFERENCES

[1] M. Banu and K. Nallaperumal. Analysis of color feature extraction techniques for pathology image retrieval systems. *International Journal of Computational Technology and Applications*, 2:1930–1938, 2010.

[2] C. Cruz, I. Ona, N. Castilla, and R. Opulencia. Bacterial blight of rice. Retrieved from http://www.knowledgebank.irri.org/decision-tools/rice-doctor/rice-doctor-fact-sheets/item/bacterial-blight.

[3] M. Gumapac. Rice: A filipino constant. *Bar Digest*, 25(10):1337–1342, 2011.

[4] L. Han, M. Haleem, and M. Taylor. A novel computer vision based approach to automatic detection and severity assessment of crop diseases. In *Science and Information Conference (SAI*, page 638–644, 2015.

[5] R. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6):1–12, 1973.

[6] S. Hosaini, S. Alirezaee, M. Ahmadi, and S.-A. Makki. Comparison of the legendre, zernike and pseudo-zernike moments for feature extraction in iris recognition. In *2013 5th International Conference on Computational Intelligence and Communication Networks*, page 483–487, 2013. IEEE. DOI: 10.1109/CICN.

[7] M. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2), 1962.

[8] T. M. Hupkens. Legendre moments of colour images. In *Netherlands Defence Academy, P.O. Box 10000, 1780 CA Den Helder*. 2009.

[9] T. Islam. *A Faster Technique on Rice Disease Detection Using Image Processing of Affected Area in Agro-Field*. Patuakhali Science and Technology University, 2018.

[10] R. Kaundal, A. Kapoor, and G. Raghava. Machine learning techniques in disease forecasting: a case study on rice blast prediction. *BMC Bioinformatics*, 7(485), 2006.

[11] N. Keen. Color moments, 2005.

[12] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. 18:1–52, 2018.

[13] S. Liao. *Image Analysis by Moments. The Department of Electrical and Computer Engineering*. The University of Manitoba, Winnipeg, Manitoba, Canada, 1993.

[14] I. Mundi. Philippines milled rice domestic consumption by year, retrieved: Jan 5, 2024, 2024.

[15] L. Nalley, F. Tsiboe, A. Durand-Morat, A. Shew, and G. Thoma. Economic and environmental impact of rice blast pathogen (magnaporthe oryzae) alleviation in the united states. *PLoS ONE*, 11(12):0167295, 2016.

[16] C. L. Novak and S. A. Shafer. Anatomy of a color histogram. *CVPR*, 92, 1992.

[17] J. W. Orillo, I. Valenzuela, and J. Cruz. Identification of diseases in rice plant (oryza sativa. In *using Back Propagation Artificial Neural Network. 10.1109/HNICEM.2014.7016248*. 2014.

[18] S. Pandey. International rice research institute rice in the global economy: Strategic research and policy issues for food security. In *ISBN: 9789712202582, LCCN: 2011308303*. 2010.

[19] J. Shah, H. Prajapati, and V. Dabhi. A survey on detection and classification of rice plant diseases. In *Department of Information Technology*, page 978–1–5090–1936–6 16 31 00, Nadiad, Gujarat, India, 2016. Dharmsinh Desai University. Eds.

[20] M. Teague. Image analysis via the general theory of moments. *J. Opt. Soc. Am*, 70(8), 1980.

[21] P. V. A high-resolution, integrated system for rice yield forecasting at district level. *Agricultural Systems*, 2018.

[22] D. Zhang, M. M. Islam, and G. Lu. A review on automatic image annotation techniques. *Pattern Recognition*, 45(1):346–362, 2012.