FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

BIT 33603
DATA MINING

IMAGE-BASED DETECTION OF RICE LEAF DISEASES BY USING DEEP CNN MODELS

PREPARED FOR:
DR. ROZITA BINTI ABDUL JALIL

PREPARED BY:
NURULAINA NISA BINTI SAHANUAN (AI220179)
NURUL JANNAH BINTI KAMARUL ZAMAN (AI220147)
SERENA NG YEN XIN (AI220061)
TUAN KHALIDAH SYAZWANA BINTI TUAN MOHD KASMAWI (AI220118)

20th JUNE 2025

IMAGE-BASED DETECTION OF RICE LEAF DISEASES BY USING DEEP CNN MODELS

Nurul Jannah Kamarul Zaman[1], Nurulaina Nisa Sahanuan[2], Serena Ng Yen Xin[3], Tuan Khalidah Syazwana Tuan Mohd Kasmawi[4] Rozita Abdul Jalil[4]*
Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia,
Parit Raja 86400, Batu Pahat, Johor, Malaysia
rozita@uthm.edu.my

Abstract. Rice is a staple food crop for over half of the global population, and its yield is frequently threatened by various leaf diseases such as Bacterial Blight (BB), Brown Spot (BS), and Leaf Smut (LS). Timely and accurate identification of these diseases is critical in mitigating crop loss. This project proposes an image-based detection using Convolutional Neural Networks (CNNs) to classify rice leaf diseases, via the Knowledge Discovery in Databases (KDD) process. By utilising a rice leaf disease dataset from Mendeley Data, which includes the three types of diseased rice leaves, CNN, Naive Bayes, Logistic Regression, and Support Vector Machine classification models are developed. The models are evaluated using performance criteria such as accuracy, precision, recall, and F1-score metrics. The results show that data mining is successful in rice leaf disease diagnosis. The outcomes of this project demonstrate the potential of deep learning models in agricultural diagnostics and precision farming.

1. Introduction

Agriculture, the main income source in many countries, relies heavily on rice as a staple food crop, particularly in Asia, where it is susceptible to various diseases at different growth stages [1]. Detecting rice leaf disease is crucial as they significantly affect yield, quality, and ultimately directly affecting the economy and food security [2]. Plant diseases, if not identified and treated in a timely manner, can lead to substantial losses, reduce grain quality, and impair market value.

Conventional methods of disease identification typically involve manual inspection by agricultural experts. While effective to some extent, this method is labour-intensive, time consuming, and highly dependent on human expertise. Moreover, visual diagnosis is prone to inconsistencies due to subjective interpretation and human error. These challenges highlight the need for automated, efficient, and accurate systems that can support farmers in disease detection and management.

With the advancement of machine learning and deep learning, image-based disease detection has emerged as a promising solution to automate and accelerate the diagnosis process of rice leaf diseases. Deep learning models such as

Convolutional Neural Networks (CNNs) have demonstrated exceptional performance in image classification, object detection, and image segmentation [3].

This project focuses on the development of a deep learning model to detect and classify rice leaf diseases based on images. The dataset utilised in this study, the Rice Leaf Disease Dataset, is sourced from Mendeley Data, contains labelled images across rice leaf diseases. The collection of data focuses on three major diseases that affect rice leaves, which are Bacterial Blight (BB), Brown Spot (BS), and Leaf Smut (LS). A deep Convolutional Neural Network (CNN) model is developed and compared with traditional classifiers such as Naive Bayes, Logistic Regression, and Support Vector Machine. By leveraging Convolutional Neural Networks (CNNs), this project aims to assist in classifying between healthy leaves and leaves affected by major rice diseases such as bacterial blight, brown spot, and leaf smut.

A Convolutional Neural Network (CNN) model is developed for this purpose and compared with traditional machine learning classifiers such as K-Nearest Neighbours (KNN) and Logistic Regression. By leveraging the hierarchical learning capability of CNNs, this project aims to accurately classify diseased and healthy rice leaves and evaluate the performance of different models using standard metrics such as accuracy, precision, recall, and F1-score. Meanwhile, traditional classifiers like Naive Bayes, Logistic Regression, and Support Vector Machine are included for comparative analysis to understand their limitations and evaluate how deep models outperform shallow ones, especially in image-based tasks.

The main objectives of this study are as follows:
1) To preprocess and analyse the dataset for rice leaf disease prediction.
2) To implement classification algorithms using the R programming language and Python programming language and evaluate their performance.
3) To compare the prediction accuracy of each algorithm and identify the most effective model based on standard evaluation metrics.

CNN is a highly influential deep learning model that has achieved remarkable success in fields like computer vision and natural language processing, garnering significant interest from both industry and academia in recent years [4]. Deep CNN is a specialised neural network renowned for its outstanding performance in computer vision and image processing tasks, achieved through multi-layered feature extraction architecture that autonomously learns hierarchical representations from data [5]. These capabilities make CNNs ideal for tasks involving disease classification based on leaf images.

The motivation behind this project is to contribute towards smart farming systems that can aid farmers in early disease detection and management. By integrating data mining techniques with domain-specific knowledge in agriculture, precision agriculture and sustainable food production can be achieved with the help of technology.

The remainder of this paper is organized as follows. Section 2 reviews all work related to rice leaf disease prediction using data mining techniques. Section 3 presents the Knowledge Discovery in Database (KDD) methodology utilised in performing the data mining task, along with the dataset and the evaluation metrics. Section 4 presents the outcomes. Section 5 discusses the research findings, and finally, Section 6 concludes with some directions for future work.

2. Related Work

Numerous studies have applied deep learning techniques to detect plant diseases, with a growing focus on rice leaf diseases in recent years. A review of seven relevant works that either used the same dataset or similar datasets, and applied CNN-based approaches for disease classification is shown in Table 1.

Table 1: Summary Table of Related Works

Reference	**Purpose of Study**	**Machine Learning Methods Used**	**Obtained Results**
K. Ahmed et al. (2019) [6]	To detect rice leaf diseases using machine learning techniques for early diagnosis and yield protection.	K-Nearest Neighbour (KNN), Decision Tree (J48), Naive Bayes, Logistic Regression	Decision tree algorithm, after 10-fold cross validation, achieved an accuracy of over 97% when applied on the test dataset.
P. K. Sethy et al. (2020) [7]	To identify rice leaf disease by leveraging deep features from CNNs and classifying them with SVM.	Convolutional Neural Network (CNN), deep features extraction + SVM approach	Deep features of ResNet50 and SVM classification model is superior among deep feature approaches.

H. Palllathadka et al. (2022) [8]	To apply ML/DL techniques and automate rice disease detection to increase crop yield.	Support Vector Machine (SVM), Naïve Bayes, Convolutional Neural Network (CNN)	Accuracy of ML algorithms: SVM – 96.2% CNN – 91.3% Naïve Bayes – 78.8%
M. F. X. Cham et al. (2021) [9]	To develop an Android-based application for mobile-compatible rice disease identification.	Convolutional Neural Network (CNN)	The level of accuracy generated from the model formed in classifying disease images on rice leaves in this study is 80%
R. Dogra et al. (2023) [10]	To detect brown spot disease in rice leaves using deep learning for smart agriculture.	CNN-Visual Geometry Group 19 (VGG19)	The developed model's highest achievement is accuracy = 93.0%, precision = 92.4%, sensitivity = 89.9%.
P. Vasavi et al. (2022) [11]	A comprehensive overview of recent research in the field of crop leaf disease prediction using image processing (IP), machine learning (ML) and deep learning (DL) techniques	K-Nearest Neighbours (KNN), Neural Networks (NN), Support Vector Machine (SVM), Naïve Bayes, Convolutional Neural Network (CNN)	The results showed that the multi-class SVM performed better than other techniques, with a precision of 98.63%.

P. Tejaswini et al. (2022) [12]	To help in detecting rice leaf diseases to get a healthy crop yield.	CNN – VGG-19, VGG-16, XCeption, ResNet50, 5-layer convolution	The 5-layer convolution model had the best accuracy of 78.2 %, while others, such as VGG16, had a lower accuracy of 58.4%.

3. Methodology

This data mining task involves detecting and classifying rice leaf diseases by image-based data and classification algorithms. The task follows the Knowledge Discovery in Databases (KDD) methodology to extract meaningful patterns from the image dataset. The flow of the KDD process is illustrated in Fig 1.

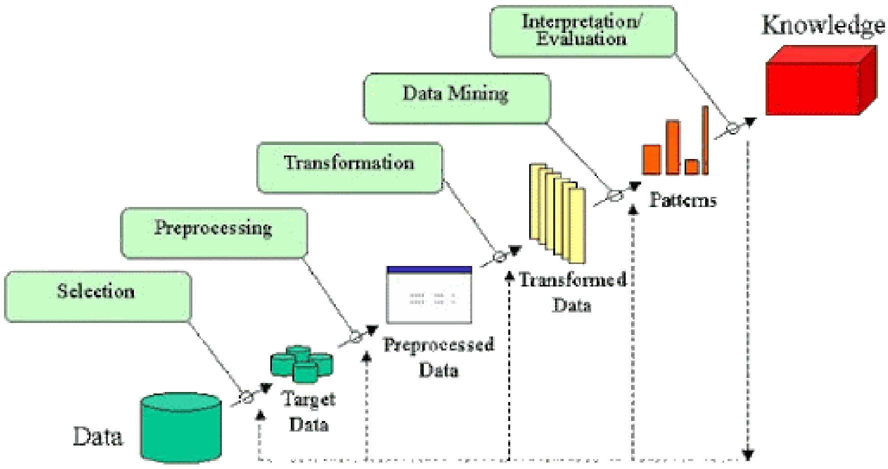

Fig. 1 Knowledge Discovery in Database (KDD) Process

In the data selection phase, a publicly available dataset titled Rice Leaf Disease Image Dataset was selected from Mendeley Data, containing a total of 4684 high-resolution images. The images belong to three categories of rice leaf diseases which are Brown Spot (BS), Bacterial Blight (BB), and Leaf Smut (LS). Each class includes a different number of images, and each image is labeled according to its respective disease. This dataset was selected due to its balanced distribution, high image quality, and relevance to the agriculture and crop disease classification domain.

During the data preprocessing stage, the images were cleaned and standardized. All images were resized to 64 X 64 pixels and used only 1460 images for each label to maintain consistency across the dataset. Image normalization was applied by augmentation techniques such as rotation, horizontal flipping, and zooming were also applied to artificially increase the database size and introduce variability, which helps prevent overfitting and improves the generalization of the models.

In the data transformation phase, the image files were converted into structured numerical arrays using image processing libraries in R through the keras and tensorFlow packages. Class labels were encoded using one-hot encoding to fit the multi-class classification structure. The data was then organized into training and testing sets using splits from 30:70, 40:60, 50:50, 60:40, and 70:30 to enable effective model training and performance evaluation.

The data mining phase involved building and training multiple classification models to detect rice leaf diseases based on image features. Three algorithms were used: Convolutional Neural Network (CNN), Naive Bayes, Logistic Regression, and SVM. The CNN model was developed using the keras interface in R and considered convolutional, pooling, and dense layers designed to automatically extract deep features from the image data. For comparison, the other algorithms like SVM, Naive Bayes, and Logistic Regression were trained using flattened image vectors to evaluate their performance on the same task.

In the interpretation/evaluation phase, the trained models were tested on the reserved test set to assess their prediction capabilities. The performance of each classifier was evaluated using standard metrics including accuracy, precision, recall, and F1-score. Visualization tools such as confusion matrices and classification reports were used to better understand the model outputs and diagnose misclassification trends. Through this evaluation, the CNN model demonstrated superior performance in accurately classifying the rice leaf diseases, confirming the effectiveness of deep learning approaches for image-based agricultural diagnostics.

All experiments were conducted in RStudio. the dataset was loaded and processed using image manipulation functions and machine learning libraries. A random seed was set to ensure the reproducibility of results. Models were trained and validated using 5-fold cross-validation, providing reliable estimates of their generalization performance.

The classification task in this project supports the goal of improving automated disease detection in agriculture, helping farmers make informed and timely decisions in crop management. This not only enhances decision-making but also empowers farmers in rural or undeserved areas where expert consultation is limited. As a result, the project contributes toward smart farming initiatives and sustainable agriculture through the application of data mining techniques.

3.1 Dataset

This dataset falls within the precision agriculture and plant pathology domains. Table 1 provides the specific details of the dataset, including its source, publishing year, owner, sample size, number of attributes, usage, missing values, and data types.

Table 1. Specific Description of the Dataset

Description	
Dataset Name	Rich Leaf Diseases Dataset
Source	Mendeley Data
Publishing Year	19 Oct 2023
Owner	Lourdu Antony, Leo Prasanth
Sample Size	4684
Number of Classes	3
Class Labels	Bacterial Blight (BB), Brown Spot (BS), and Leaf Smut (LS)
Image Format	JPG
Image per Class	Bacterial Blight (1604), Brown Spot (1620), and Leaf Smut (1460)

The dataset shows that each class has a different number of images. To increase the efficiency, only 1460 images are chosen from each class and the excessive data will be deleted. Table 2 shows the dataset's class labels with the number of images that are used.

Table 2. Class Labels and Description

Class Labels	**Description**	**Number of Images**
Bacterial Blight (BB)	Yellowish lesions from the tip downward, water-soaked	1460
Brown Spot (BS)	Round to oval brown lesions are scattered on the leaf	1460
Leaf Smut (LS)	Narrow, black streaks or smudges along the leaf	1460

3.2 Algorithms

- Convolutional Neural Network (CNN)

  Convolutional Neural Network is a deep learning algorithm used in image processing and classification, inspired by the human brain's visual cortex. CNN automatically learns spatial hierarchies of features using backpropagation with multiple building blocks such as convolution layers, pooling layers, and fully connected layers.

  The formula of Convolutional Neural Network (CNN) is shown in Eq. 1.

$$Z = (X * W) + b \tag{1}$$

  where Z = output feature map, X = input image matrix, W = convolutional filter (weights), and b = bias.

  The implementation of a convolutional neural Network (CNN) in this experiment using Python is shown in Fig. 2.

```
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(64, 64, 3)),
    MaxPooling2D(pool_size=(2, 2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dense(num_classes, activation='softmax')
])

model.compile(optimizer=Adam(), loss='categorical_crossentropy', metrics=['accuracy'])

model.fit(train_data, steps_per_epoch=5, epochs=3, validation_data=test_data, validation_steps=2)
```

Fig. 2 Implementation of Convolutional Neural Network (CNN) in Python

- Naive Bayes

  Naive Bayes is a Bayes' Theorem-based probabilistic classification assuming that predictors are independent of each other. Naive Bayes does well with high-dimensional data and is simple to implement.

  The formula of Naive Bayes is shown in Eq. 2.

$$p(c \mid x) = \frac{p(x \mid c).p(c)}{p(x)} \tag{2}$$

  where p(c|x) = posterior probability, p(x|c) = likelihood, p(c) = prior probability, and p(x) = evidence

The implementation of Naive Bayes in this experiment using Python is shown in Fig. 3.

```
model = GaussianNB()
model.fit(X_train, y_train_enc)
y_pred = model.predict(X_test)
```

Fig. 3 Implementation of Naive Bayes in Python

- Logistic Regression

Logistic regression is a statistical analysis technique used to describe the relationship between one or more independent predictor factors and a binary outcome, or a dependent variable with two categories, yes or no.

The formula of logistic regression is shown in Eq. 3.

$$p(x) = \frac{e^{a+bx}}{1+e^{a+bx}} \tag{3}$$

where p(x) = predicted output, a = intercept term, and b = coefficient of single input value (x).

The implementation of logistic regression in this experiment using Python is shown in Fig. 2.

```
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

Fig. 4 Implementation of Logistic Regression in Python

- Support Vector Machine (SVM)

Support Vector Machine is a supervised machine learning technique that seeks the best hyperplane to divide data points of various classes. It works very well with high-dimensional image data.

The formula of Support Vector Machine (SVM) is shown in Eq. 4.

$$f(x) = w^T x + b \tag{4}$$

Subject to the constraint in Eq. 5:

$$y_i(w^T x_i + b) \geq 1 \tag{5}$$

where wT = transpose of weight vector, x□ = feature vector, y□ = class label, and b = bias.

The implementation of a support vector machine (SVM) in this experiment using Python is shown in Fig. 2.

```
# Train SVM
model = SVC(kernel='linear')  # You can change to 'rbf' or 'poly' if needed
model.fit(X_train, y_train_enc)

# Predict
y_pred = model.predict(X_test)
```

Fig. 5 Implementation of Support Vector Machine (SVM) in Python

3.3 Evaluation Metrics

The evaluation metrics used in the experiments are accuracy, precision, recall, and F-score values.

- Accuracy. Accuracy evaluates how effectively a model classifies instances into their correct categories. The formula for calculating accuracy is shown in Eq. 6.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

The metrics evaluation of accuracy in this experiment using Python is shown in Fig. 6.

```
accuracy = accuracy_score(y_test_enc, y_pred)
```

Fig. 6 Implementation of Accuracy Evaluation Metrics in Python

- Precision. Precision computes the ratio of true positive predictions (TP) to the total positive predictions (TP+FP). The formula for calculating precision is shown in Eq. 7.

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

The metrics evaluation of precision in this experiment using Python is shown in Fig. 7.

```
precision = precision_score(y_test_enc, y_pred, average='macro', zero_division=1)
```

Fig. 7 Implementation of Precision Evaluation Metrics in Python

- Recall. Recall, also known as the true-positive rate or sensitivity, calculates the probability of the model correctly identifying true positive cases. The formula to calculate recall is shown in Eq. 8.

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

The metrics evaluation of recall in this experiment using Python is shown in Fig. 8.

```
recall = recall_score(y_test_enc, y_pred, average='macro')
```

Fig. 8 Implementation of Recall Evaluation Metrics in Python

- F1-score. The F1-score represents the harmonic mean of precision and recall, offering a balanced evaluation score. The formula to calculate F1-score is shown in Eq. 9.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{9}$$

The metrics evaluation of recall in this experiment using Python is shown in Fig. 9.

```
f1 = f1_score(y_test_enc, y_pred, average='macro')
```

Fig. 9 Implementation of F1-Score Evaluation Metrics in Python

4. Results

The implemented models follow a standard machine learning pipeline: image preprocessing, feature extraction, model training, evaluation, and performance comparison. All algorithms were applied to the same dataset using different train-test split ratios (30–70, 40–60, 50–50, 60–40, and 70–30) to observe how the amount of training data impacts each model's performance.

The purpose of the experiments is to compare the performance of four classification algorithms—Convolutional Neural Network (CNN), Naive Bayes, Logistic Regression, and Support Vector Machine (SVM)—in classifying rice leaf diseases from image data. The dataset contains three disease categories: Bacterial Blight, Leaf Smut, and Brown Spot. The models were evaluated based on error rate, accuracy, precision, recall, and F1-score. The results are shown in Table 2.

Table 2. Experimental results

Data split (%)	Algorithm	Error	Accuracy	Precision	Recall	F1-score
30-70	CNN	43.68	56.32	0.69	0.55	0.46
	Naive Bayes	38.59	61.41	0.61	0.62	0.61
	Logistic Regression	18.76	81.24	0.81	0.81	0.81
	SVM	0.00	100.00	1.00	1.00	1.00
40-60	CNN	36.12	63.88	0.69	0.65	0.64
	Naive Bayes	38.59	61.41	0.61	0.62	0.61
	Logistic Regression	13.15	86.85	0.87	0.87	0.87
	SVM	00.00	100.00	1.00	1.00	1.00
50-50	CNN	43.59	56.41	0.71	0.56	0.45
	Naive Bayes	38.59	61.41	0.61	0.62	0.61
	Logistic Regression	9.50	90.50	0.91	0.91	0.91
	SVM	00.00	100.00	1.00	1.00	1.00
60-40	CNN	48.56	51.44	0.77	0.50	0.41
	Naive Bayes	38.59	61.41	0.61	0.62	0.61
	Logistic Regression	6.63	93.37	0.93	0.93	0.93
	SVM	00.00	100.00	1.00	1.00	1.00
70-30	CNN	42.13	57.87	0.72	0.57	0.46
	Naive Bayes	39.34	60.66	0.60	0.61	0.60
	Logistic Regression	4.64	95.36	0.95	0.95	0.95
	SVM	5.29	94.71	0.95	0.95	0.95

6. Discussion

The experiment involved four machine learning algorithms—CNN, Naive Bayes, Logistic Regression, and SVM—each evaluated using five different data splits. These splits (30-70, 40-60, 50-50, 60-40, and 70-30) were selected to understand the effect of training data size on classification performance. Each algorithm's performance was assessed based on error rate, accuracy, precision, recall, and F1-score.

Among all algorithms, Support Vector Machine (SVM) consistently delivered the best results, achieving 100% accuracy in almost all data splits. Even at the 70-30 split, where most algorithms perform less effectively due to reduced training data, SVM maintained a high accuracy of 94.71%. This suggests that SVM's ability to find optimal decision boundaries in high-dimensional spaces makes it highly effective for image-based classification tasks like this.

Logistic Regression also performed well, with accuracy ranging from 81.24% to 95.36% as the amount of training data increased. Its strong performance across splits shows that even a relatively simple linear model can effectively handle image features when properly preprocessed. Its precision, recall, and F1-scores remained consistently high, making it a reliable alternative to more complex models.

Naive Bayes, on the other hand, yielded moderate and consistent performance, with accuracy hovering around 61% across all data splits. This algorithm assumes feature independence, which is not realistic for image data where pixel relationships and spatial patterns are crucial. As a result, Naive Bayes struggled to model the complexity of leaf disease patterns effectively.

Surprisingly, the Convolutional Neural Network (CNN) performed below expectations despite being designed for image classification. Accuracy values for CNN ranged from only 51.44% to 63.88%, and its F1-scores remained below 0.5 in most cases. The limited performance may be attributed to the simple architecture used (with only one convolutional layer) and the relatively small size of the dataset, which can lead to underfitting or insufficient learning.

The difference in performance between the models can also be attributed to how well they handle continuous feature data from images. While SVM and Logistic Regression thrive in high-dimensional settings, Naive Bayes and the basic CNN may require either more data or deeper structures to perform equally well. Additionally, the consistent performance of SVM suggests that the disease classes in the dataset are linearly separable in the feature space created by the pixel values.

7. Conclusions and Future Work

This study focused on classifying three rice leaf diseases—Bacterial Blight, Leaf Smut, and Brown Spot—using image-based machine learning techniques. Four algorithms (CNN, Naive Bayes, Logistic Regression, and SVM) were tested across multiple data splits to evaluate their classification performance. The objective was to identify the most effective model for automatic disease detection based on standard evaluation metrics.

The results showed that SVM outperformed the other algorithms, consistently achieving the highest accuracy, precision, recall, and F1-scores. Logistic Regression also demonstrated strong performance, particularly as the training data increased. Meanwhile, Naive Bayes showed limitations due to its simplistic assumptions, and CNN's performance was restricted by its basic architecture and insufficient training data.

For future work, improving CNN performance through data augmentation, deeper architectures, or using transfer learning from pre-trained models (such as VGG16 or ResNet50) is highly recommended. Additionally, deploying these models into a real-time mobile application could make early rice disease detection more accessible to farmers. Finally, expanding the dataset and testing ensemble methods may lead to even more accurate and robust classification results in agricultural applications.

Acknowledgement. This research is supported by Universiti Tun Hussein Onn Malaysia.

References

1. Kaur, A., Guleria, K., & Trivedi, N. K. (2021, October). Rice leaf disease detection: A review. In 2021 6th International Conference on Signal Processing, Computing and Control (ISPCC) (pp. 418-422). IEEE.
2. Tejaswini, P., Singh, P., Ramchandani, M., Rathore, Y. K., & Janghel, R. R. (2022, June). Rice leaf disease classification using CNN. In IOP Conference Series: Earth and Environmental Science (Vol. 1032, No. 1, p. 012017). IOP Publishing.
3. He, Y., & Xiao, L. (2023). Structured pruning for deep convolutional neural networks: A survey. IEEE transactions on pattern analysis and machine intelligence, 46(5), 2900-2919.
4. Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects. IEEE transactions on neural networks and learning systems, 33(12), 6999-7019.
5. Khan, A., Sohail, A., Zahoora, U. et al. A survey of the recent architectures of deep convolutional neural networks. Artif Intell Rev 53, 5455–5516 (2020). https://doi.org/10.1007/s10462-020-09825-6
6. Ahmed, K., Shahidi, T. R., Alam, S. M. I., & Momen, S. (2019, December). Rice leaf disease detection using machine learning techniques. In 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI) (pp. 1-5). IEEE.

7. Sethy, P. K., Barpanda, N. K., Rath, A. K., & Behera, S. K. (2020). Deep feature based rice leaf disease identification using support vector machine. Computers and Electronics in Agriculture, 175, 105527.
8. Pallathadka, H., Ravipati, P., Sajja, G. S., Phasinam, K., Kassanuk, T., Sanchez, D. T., & Prabhu, P. (2022). Application of machine learning techniques in rice leaf disease detection. Materials Today: Proceedings, 51, 2277-2280.
9. Cham, M. F. X., Tanone, R., & Riadi, H. A. T. (2021, September). Identification of rice leaf disease using convolutional neural network based on android mobile platform. In 2021 2nd International Conference on Innovative and Creative Information Technology (ICITech) (pp. 140-144). IEEE.
10. Dogra, R., Rani, S., Singh, A., Albahar, M. A., Barrera, A. E., & Alkhayyat, A. (2023). Deep learning model for detection of brown spot rice leaf disease with smart agriculture. Computers and Electrical Engineering, 109, 108659.
11. Vasavi, P.; Punitha, A.; Rao, T.V.N. Crop leaf disease detection and classification using machine learning and deep learning algorithms by visual symptoms: A review. Int. J. Electr. Comput. Eng. (IJECE) 2022, 12, 2079–2086.