Integrated Deep Learning Framework for Automated Rice Leaf Disease Image Classification

Aman Maurya
Computer Science & Engineering
NIT Rourkela
Odisha, India
223cs3153@nitrkl.ac.in

Dr. Sibarama Panigrahi
Computer Science & Engineering
NIT Rourkela
Odisha, India
email@nitrkl.ac.in

***Abstract*—In the field of machine learning, feature extraction is important for the accurate classification of images. Building on the advancements in Convolutional Neural Networks (CNNs), this study addresses the identification of rice leaf diseases by leveraging both deep and handcrafted features to enhance classification performance. Utilizing dataset comprising 5932 on field images of rice leaves affected by bacterial blight, brown spot, tungro, and blast, we checked the efficacy transfer learning and also of combining deep features from multiple CNN models together and with handcrafted features. The performance of the hybrid approaches was benchmarked using multiple shallow machine learning classifiers. Results indicated that while transfer learning's performance for ResNet50, VGG16 and VGG19 were good, the results from feature fusion between VGG16 and VGG19 gave the best overall result. The performance of Handcrafted Features specifically LBP with shallow machine learning classifiers was comparable with transfer learning. The best performance was denoted by ResNet50 using transfer learning approach, deep feature of VGG16 with SVM classifier, feature fusion of VGG16 + VGG19 using SVM classifier and ResNet50 + EfficienteNetB3, EfficientNetB3 + InceptionV3 with XGBoost Classifier, and Our hybrid models VGG16+LBP, EfficientNetB3+LBP outperformed traditional CNN-SVM models. Finally the combination of VGG19+ResNet50+LBP showed that this type of combination outperforms every previous approach. Additionally, this approach surpassed other traditional image classification methods such as bag-of-features, local binary patterns (LBP) plus SVM, and the Gray Level Co-occurrence Matrix (GLCM) plus SVM. This study underscores the potential of integrating deep and handcrafted features for robust image classification in agricultural applications, offering a promising solution for the early and accurate detection of rice leaves diseases, ultimately aiding in effective crop management and disease control.**

***Index Terms*—Transfer-Learning, Deep-Feature-Fusion, Handcrafted-Features, VGG16, VGG19, ResNet50, HOG, LBP, Deep-Learning, Feature-Extraction, Image-Classification, Rice-leaf-disease**

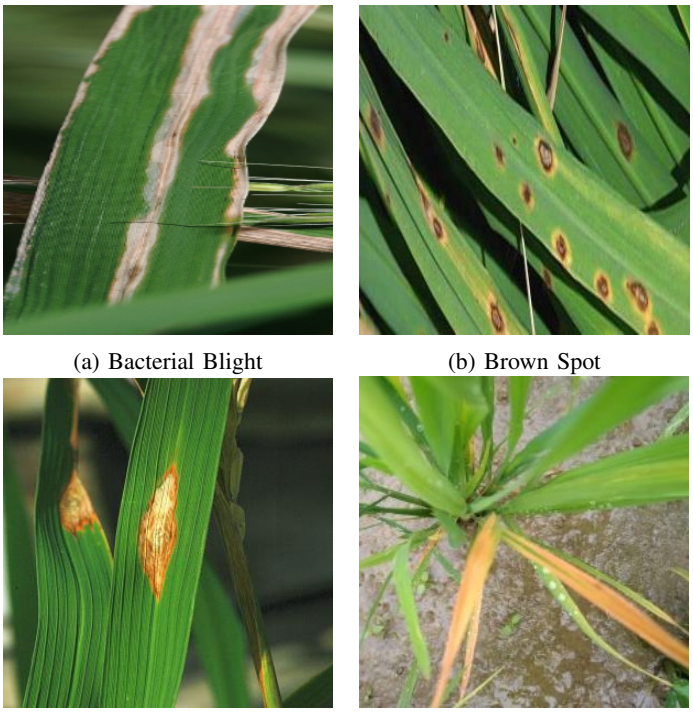
I. Introduction

Rice is one of main food sources in India, with land under rice cultivation more than even China's cultivation. Odisha ranks fourth among Indian states in rice production, with the western area, specially the Sambalpur and Bargarh districts which are known as the rice bowl of Odisha, are renowned for rice cultivation. This region cultivates various rice varieties across two farming seasons, Kharif season (which is from July to October), dependent on the monsoon, and Rabi season (which is from October to March), reliant on the Hirakud dam's water supply. Each year, paddy fields suffer from various diseases and pest attacks, posing significant challenges to local farmers. Particularly, young farmers with less experience in agriculture struggle to identify these diseases, leading to ineffective pesticide application. This pressing issue motivated our research on identifying rice diseases prevalent in western Odisha. In this region, four primary rice diseases are commonly observed: bacterial blight, brown spot, tungro and blast. The images of these diseases are listed in Figure 1. Early detection of these diseases is of great importance for full-blown disease prevention and plant treatment at a later stage. It also plays a vital role in the management and decision-making of agricultural production (Liu et al., 2018). Disease-infected plants tend to show obvious marks or lesions on leaves, flowers or fruits. Generally, each disease presents a unique visible pattern that can be used to diagnose plant abnormalities. The leaves of plants are the primary source for identifying plant diseases, and most of the symptoms of diseases appear on the leaves (Ebrahimi et al., 2017).

Traditional methods of identifying diseases, such as visual inspection and laboratory experimentation, are either time-consuming or require expert personnel and chemical reagents, making them impractical for widespread use. While mobile applications like Rice Doctor app and Rice Xpert app have been developed to assist farmers, they often yield inaccurate diagnoses and lack efficiency. Numerous research studies have focused on automatic rice disease diagnosis using machine learning techniques and image processing techniques. These studies employ various methods, including support vector machines, pattern recognition, computer vision and digital image processing and, not only for rice, but also for other crops.

However, traditional machine learning methods have limitations, such as limited data handling capabilities and the requirement for segmentation and feature extraction. Deep learning techniques, particularly deep convolutional neural networks (CNNs) and transfer learning have shown promise in overcoming these limitations by handling large datasets and eliminating the need for pre-processing steps. Further, extraction of deep features and classification using shallow machine learning classifiers takes less time and is equally or more accurate than transfer learning technique. Although

deep learning methods have been shown to be very capable in depicting both high-level and low-level features, they are less reliable than handcrafted features in representing local spatial characteristic (Cai et al. 2018). Thus, to better capture local characteristics that exclusively exist in plant leaf image, we have tried to fuse the handcrafted and deep features, with the handcrafted features complementing the deep features.



(a) Bacterial Blight (b) Brown Spot

(c) Blast (d) Tungro

Fig. 1: Rice Leaf Disease Image

II. Literature Review

In recent years, numerous research papers have been published on automated rice disease diagnosis using image processing, machine learning and deep learning techniques. Methods based on deep learning have significantly progressed in the field of computer vision (Huang et al., 2017; Chen et al., 2017; Krizhevsky et al., 2012; Simonyan and Zisserman, 2014; Szegedy et al., 2015; He et al., 2016; Ye et al., 2019; Chen and Yun, 2021; Badrinarayanan et al., 2017; Wang et al., 2021). These techniques, known for their ability to extract meaningful feature representations, have also been applied to plant disease recognition and detection (Mohanty et al., 2016; Sladojevic et al., 2016; Nachtigall et al., 2016; Jalal and Burak, 2020; Bi et al., 2020; Shrivastava and Pradhan, 2021; Shrivastava et al., 2021).

Sladojevic et al. (2016) proposed the first deep learning-based model for plant disease classification using deep convolutional networks. Mohanty et al. (2016) presented a Convolutional Neural Network (CNN) based model for detecting 26 diseases across 14 crop species, achieving promising results. Yu et al. (2020) introduced an attention mechanism to highlight leaf areas, enhancing the capture of discriminative features in diseased leaves. Tetila et al. (2019) utilized data augmentation and fine-tuning to train a deep network model for automatic identification of soybean leaf diseases. Yu et al. (2020) also developed a region-of-interest based two-stream network for recognizing apple leaf diseases, focusing on diseased leaf areas while separating the background. Jalal et al. (2020) applied a deep neural network (DNN) to detect apple leaf diseases, using SURF for feature extraction and an evolutionary algorithm for feature optimization to create a plant disease detection system. Bi et al. (2020) proposed a leaf disease model based on the MobileNet architecture and compared it with ResNet152 and InceptionV3 models.

However, recognizing leaf diseases remains challenging due to two key factors: 1) the small and variable size of disease spots, and 2) issues like complex backgrounds and uneven lighting, which result in low inter-class and high intra-class variation of diseased samples.

This study employed a variety of deep learning architectures, namely VGG16, VGG19, ResNet50, EfficientNetB3, and InceptionV3 and handcrafted features namely HOG, LBP, SIFT, GLCM and Gabor Filters. Each architecture was chosen and combined for its ability to extract deep features from the dataset. The methods applied encompassed deep feature extraction, transfer learning to enhance model performance, and comprehensive evaluation metrics such as Accuracy Score, Precision, Recall (Sensitivity), and F1 Score. The main goal was to develop a robust framework for improving the recognition of plant diseases using deep learning techniques. This involved observing transfer learning and integrating deep learning features with handcrafted features through feature fusion.. Handcrafted features captured detailed local information that complemented the deep learning features, facilitating the extraction of distinctive texture and shape patterns in diseased leaves. Moreover, this approach enabled the extraction of high-level semantic information relevant to different disease types. The study aimed to leverage these methods to achieve more accurate and reliable disease classification in agricultural settings.

A. Contributions

- Used transfer learning technique to train top 2 layers of a pre-trained CNN model to evaluate it's performance of rice leaf disease dataset.
- Utilized deep features extracted from various CNN models with 5 different machine learning classifiers for rice disease classification.
- Validated the approach using on-field images, enhancing upon traditional offline methods.
- Conducted a comparative analysis using SVM, Random Forest, Decision Tree, KNN and XGBoost classifiers on deep features extracted from single CNN Model, fusion of deep features extracted from pair of CNN Models, extracted handcrafted features (SIFT, GLCM, HOG, etc) and fusion of deep features of a CNN Model and Handcrafted Features together.

TABLE I: Distribution of Rice Leaf Disease Samples

Leaf Diseases	Total images count	Train and Validation Images count	Test images count
Brown Spot	1600	1400	200
Blast	1440	1240	200
Tungro	1308	1108	200
Bacterial Blight	1584	1384	200
Total	**5932**	**5132**	**800**

III. Materials and Methodology

A. Dataset

5932 photos of sick rice leaves, including brown spot, bacterial blight, tungro and blast. types, are included in the dataset. First, a high-resolution photo of a variety of rice fields in west part of Odisha was taken with a Nikon DSLR-D5600 and an 18–55 mm lens. The original big photos were used to remove the sick area patches. A picture library of agricultural pests and insect pests was used for some photos of rice illnesses.

Each patch was scaled to 300 by 300 pixels and used as a data sample. The four types of rice leaf diseases are depicted in Fig. 1. 800 photos total—200 for each category—were taken from the original dataset and saved for testing. The dataset was improved by augmentation. Simple picture rotations and flipping operations, such as rotating an image 90 degrees to the right or left, flipping it vertically or horizontally, and rotating it 180 degrees, were applied to every image as part of the augmentation process. The name and quantity of the photos utilised for the experiment are listed in Table 1. 80:20 proportions of the data samples are randomly selected for training and validation, respectively. For every execution, a random selection of training and validation samples is made for robustness.

B. Methodology

Getting and processing raw pictures of rice leaves are the first steps in the process. We organised the picture data into directories and linked the disease categories to number labels using a vocabulary. All images were reduced in size to 224 by 224 pixels, the standard size of input for a convolutional neural network. After being read in BGR format, the images were converted to RGB and scaled properly. This preparation stage was applied to the training and testing datasets. Following processing, the images were converted into NumPy arrays, the format required for input into neural network models. To determine the best technique for accurately classifying rice leaf illnesses, we have used following five distinct approaches to observe various combinations of different image classification algorithms and compared their outcomes.

*1) **Transfer Learning***

We used a technique called transfer learning, which fine-tunes pre-trained models using a huge dataset (ImageNet) for a particular job. Several picture identification tasks benefit from this method, which makes use of the learnt properties of these models. Five pre-trained models were utilised: InceptionV3, ResNet50, EfficientNetB3, VGG16, and VGG19. We excluded the top layers of each model, which are unique to the ImageNet classification job, and initialised each model with the ImageNet weights. The model was subsequently modified with the addition of unique dense layers to address our four-class categorization issue. The models were assembled with the help of the typical options for multi-class classification tasks: the Adam optimizer and the categorical cross-entropy loss function. We used the training data to train each model, reserving some data for validation. The model learnt to minimise the loss function during the course of several training epochs. This method's block diagram is displayed in Figure 2.

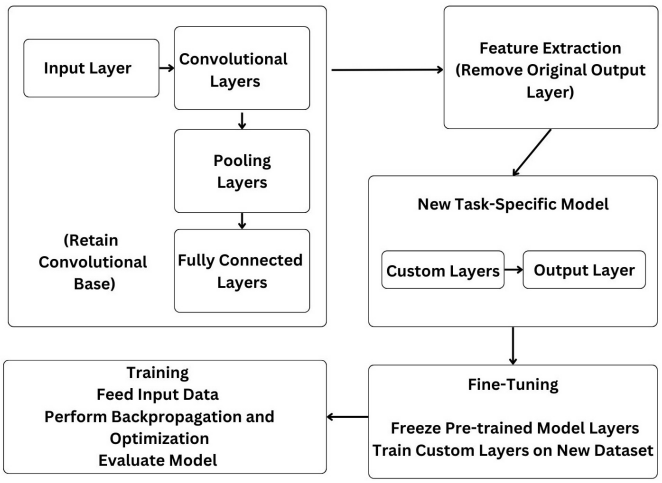


Fig. 2: Transfer Learning Process

We assessed each model's performance on the test dataset after it had been trained. Recall, Accuracy, precision, F1 score, and AUC (Area Under the ROC Curve) were among the measures we employed for assessment. These measures offer a thorough understanding of the model's performance, encompassing not only total accuracy but also the model's capacity to distinguish between classes and the trade-off between precision and recall.

We ran ten training and evaluation runs for each model to make sure the findings were reliable and robust. This made it easier to take training variability into account and gave a more realistic evaluation of each model's performance. To determine the consistency and stability of the models, we kept track of the

metrics for every run and computed the average and standard deviation.

2) Deep Feature Extraction & Shallow Machine Learning Classifier

We choose the following five pre-trained CNN models: InceptionV3, ResNet50, EfficientNetB3, VGG16, and VGG19. The top layers of the models, which are particular to ImageNet classification, were removed, and initial weights pre-trained on the ImageNet dataset were used. We employed these models as feature extractors rather than immediately applying them to classification tasks. To get a compact feature representation for every image, we combined the output of these models with a global average pooling layer. We extracted the global average pooling layer's output from each model after running the photos through it to create feature representations for the training and testing datasets. The input for the next round of categorization was these features, which contained high-level representations of the images.

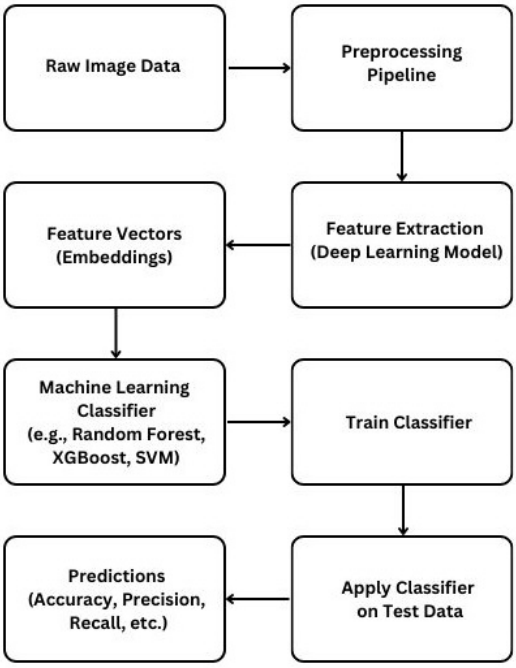


Fig. 3: Feature Extraction and Using Classifier

We evaluated the collected characteristics using a variety of machine learning algorithms in the classification step. Random Forest, Support Vector Machine (SVM), K-Nearest Neighbours (KNN), Decision Tree, and XGBoost were the classifiers that were employed. The features taken from the test set were used to evaluate each classifier, and the features from the training set were used to train them. Figure 3 shows block diagram of this process.

To fully evaluate the model's performance, we calculated a number of performance indicators, such as recall, accuracy, precision, F1 score, and AUC (Area Under the ROC Curve). 10 runs of the evaluation were performed to guarantee the reliability and consistency of the findings. To comprehend the variability and stability of the models, we logged the metrics for each run and computed the average and standard deviation.

3) Deep Feature Fusion and Shallow Machine Learning Classifier

Similar to the preceding step, feature extraction from five models was carried out. We merged features from various pairs of the five CNN models to improve performance. VGG16+VGG19, VGG19+ResNet50, ResNet50+EfficientNetB3, ResNet50+InceptionV3, and EfficientNetB3+InceptionV3 were the pairs that were observed. We used both of the models in each pair to extract features from the photos, and then we concatenated these features to create an extensive feature set. The goal of this strategy was to use the complementary qualities of several models to capture a wider range of properties.

We used the following five machine learning classifiers in the classification phase: Random Forest, Support Vector Machine (SVM), K-Nearest Neighbours (KNN), Decision Tree, and XGBoost. These classifiers were used to assess the combined features. The test dataset was used to evaluate each classifier once it had been trained using the merged feature sets. In this stage, the classifiers were fitted to the extracted features, and their performance was evaluated using a range of measures, including F1 score, accuracy, recall, precision, and AUC (Area Under the ROC Curve).

We ran 10 runs of each model-classifier combination to make sure our results were reliable and robust. We documented the performance measures for every run and computed the average and standard deviation to provide light on the models' stability and variability.

4) Handcrafted Feature Extraction and Shallow Machine Learning Classifier

In this method, first we processed the dataset, which included resizing all images to a standard dimension of 224x224 pixels and converting them to grayscale where necessary. The grayscale conversion was essential for many feature extraction techniques that rely on intensity values rather than color information. We employed five different handcrafted feature extraction techniques: Local Binary Patterns (LBP), Histogram of Oriented Gradients (HOG), Scale-Invariant Feature Transform (SIFT), Gray-Level Co-occurrence Matrix (GLCM), and Gabor features. Each technique captures different aspects of the image data.

LBP captures local texture information by comparing each pixel to its surrounding neighbors. We computed LBP features with a radius of 3 and 24 sampling points, followed by normalizing the resulting histograms. HOG focuses on the gradient orientation in localized portions of an image. HOG features are effective for capturing edge information and object shapes. SIFT is a robust feature descriptor that identifies and describes local features in images, invariant to scale and rotation. GLCM calculates the frequency of different combinations of pixel brightness values (co-occurrences) at a given offset,

capturing texture information related to the spatial distribution of pixels. Gabor Features are derived from Gabor filters, which are effective for texture representation and discrimination, capturing both spatial and frequency information.

After extracting features using each technique, we evaluated their classification performance using five different machine learning classifiers: Random Forest, Decision Tree, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and XGBoost. Each classifier was trained and tested on the feature sets derived from the respective extraction techniques. Each feature extraction method was evaluated in combination with each classifier over multiple runs to ensure the reliability of the results. We conducted 10 runs for each combination, recording metrics such as recall, accuracy, precision, F1 score, and AUC (Area Under the ROC Curve). This repetition allowed us to compute the mean and standard deviation for each metric, providing a comprehensive assessment of performance variability and robustness.

*5) **Fusion of Handcrafted Features with Deep Features***

Handcrafted Features were employed in conjunction with an extensive procedure to extract and evaluate features, utilising an array of deep learning models and machine learning classifiers. The plan was to combine the discriminative power of handcrafted features with the rich feature representations that state-of-the-art convolutional neural networks (CNNs) had learned for texture analysis. Initially, we employed the four CNN architectures (ResNet50, EfficientNetB3, VGG16, and VGG19) that showed good performance in the feature extraction method. The models were pre-trained on the ImageNet dataset, which allowed them to extract hierarchical characteristics from the images. Every model was configured to employ Global Average Pooling to obtain deep features after freezing its convolutional layers to maintain its learned representations.

After extracting features from each CNN model, we included Local Binary Pattern (LBP), the handcrafted feature that performed the best in the prior procedure, as an extra feature descriptor. In addition to providing supplementary texture information to the CNN-derived features, LBP computes texture descriptors that capture local patterns within grayscale images. By augmenting the feature space with texture-based insights in addition to the high-level visual representations that the CNNs had learnt, this fusion attempted to improve classification performance. We used a variety of machine learning classifiers for the classification tasks, including XGBoost, Random Forest, Decision Tree, Support Vector Machine (SVM), and K-Nearest Neighbours (KNN). The concatenated feature vectors containing the texture-based features from LBP and the deep features from CNN models were used to train each classifier. Using this strategy, I was able to investigate a variety of learning algorithms for efficiently classifying photos into distinct categories, including instance-based learning, ensemble methods, and gradient boosting.

In order to train machine learning classifiers to observe the performance of combining multiple deep features and handmade features together, the highest performing pair in the fusion of extracted deep features : VGG19+ResNet50 was also concatenated with the LBP features. The process is depicted in Figure 4.

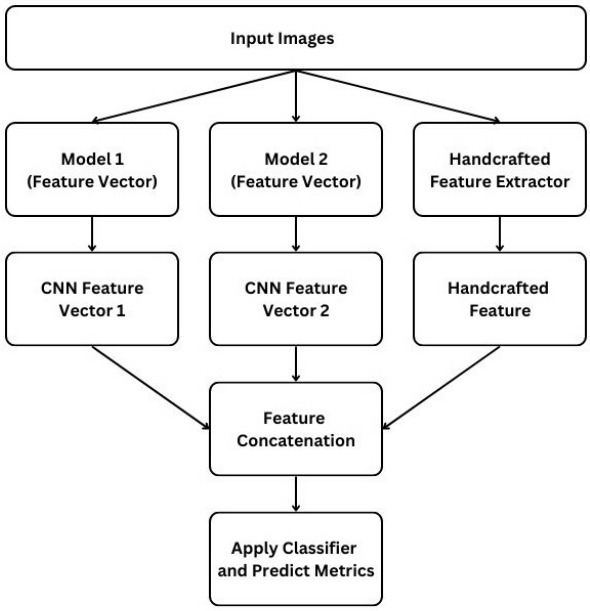


Fig. 4: Fusion of Deep features with Handcrafted features

We carried out ten iterations of the classification experiments to guarantee the results' stability and dependability. We were able to calculate average performance metrics, including F1 score, accuracy, recall, precision, and area under the curve (AUC), as a result.

IV. Simulation Results

A. Experimental Setup

In this study, we examined the performance of classification models for rice leaf disease identification using the above mentioned five type of methods. The experimental studies were implemented using Jupyter notebooks on top of Visual Studio Code, utilizing the TensorFlow package. All applications ran on a MacBook Air with an Apple M1 chip, 16GB unified memory, 256GB SSD, running macOS, and integrated Apple GPU. The batch size was 5, optimizer used was 'Adam' and loss function used was 'Categorical Crossentropy'. For each dataset, we randomly allocated 80% samples for training, and the remaining 20% data are used for testing.

B. Performance Measures

The performance of each classifier was evaluated in terms of recall, accuracy, precision, F1 Score, and Area under curve (AUC). The comparison of each classifier's performance is discussed in the following subsections.

The confusion matrix for a classifier with 4 classes can be represented as follows:

TABLE II: Results of Classification using Transfer Learning Method

Measures	**VGG16**	**VGG19**	**RESNET50**	**EFFICIENTNETB3**	**INCEPTIONV3**
Accuracy	0.994 ± 0.004	0.995 ± 0.007	**1.000 ± 0.000**	0.997 ± 0.004	0.772 ± 0.031
Recall	0.994 ± 0.004	0.995 ± 0.007	**1.000 ± 0.000**	0.997 ± 0.004	0.772 ± 0.031
Precision	0.994 ± 0.004	0.995 ± 0.007	**1.000 ± 0.000**	0.997 ± 0.004	0.779 ± 0.022
F1 Score	0.994 ± 0.004	0.995 ± 0.007	**1.000 ± 0.000**	0.997 ± 0.004	0.768 ± 0.032
AUC	0.999 ± 0.000	0.999 ± 0.000	**1.000 ± 0.000**	**1.000 ± 0.000**	0.942 ± 0.009

Highlited values show the best performance

Predicted	Class 1	Class 2	Class 3	Class 4
Actual Class 1	TP_1	$FN_{1\to2}$	$FN_{1\to3}$	$FN_{1\to4}$
Actual Class 2	$FN_{2\to1}$	TP_2	$FN_{2\to3}$	$FN_{2\to4}$
Actual Class 3	$FN_{3\to1}$	$FN_{3\to2}$	TP_3	$FN_{3\to4}$
Actual Class 4	$FN_{4\to1}$	$FN_{4\to2}$	$FN_{4\to3}$	TP_4

Where:

- **TP (True Positive)**: Correctly predicted instances for the corresponding class.
- **FN (False Negative)**: Instances of the actual class incorrectly predicted as other classes.

Performance Metrics

The performance of classifiers is measured using the following metrics:

- **Accuracy**:

$$\text{Accuracy} = \frac{\sum_{i=1}^{4} TP_i}{\sum_{i=1}^{4}(TP_i + \sum_{j=1}^{4} FN_{i\to j})}$$

- **Sensitivity (Recall) for each class**:

$$\text{Sensitivity (Class } i) = \frac{TP_i}{TP_i + \sum_{j=1}^{4} FN_{i\to j}}$$

- **Precision (Class** i):

$$\text{Precision (Class } i) = \frac{TP_i}{TP_i + \sum_{j=1}^{4} FN_{j\to i}}$$

- **Area Under the ROC Curve (AUC)**:

$$\text{AUC} = \int_0^1 TP(FP)\, d(FP)$$

- **F1 Score (Class** i):

$$\text{F1 Score (Class } i) = \frac{2 \cdot \text{Sensitivity (Class } i) \cdot \text{Precision (Class } i)}{\text{Sensitivity (Class } i) + \text{Precision (Class } i)}$$

C. Result of Transfer Learning Method

Table II summarizes the performance metrics of various models evaluated for the classification of rice leaf diseases using transfer learning technique. Here are the key insights from the results:

a. **VGG16**, **VGG19**, and **EfficientNetB3** achieved similar accuracies of around 99.5%. Their test accuracies, recall, precision and other metrics were consistently high, indicating robust generalization capabilities.
b. **InceptionV3** had all the metrics ranging from 76.7% to 77.93% which indicates it's comparably poorer performance than other models.
c. **ResNet50** demonstrated perfect score across all metrics (1.0), indicating strong performance in both training and testing phases.

Overall, the results depict the suitability of deep learning models like ResNet50, VGG16, VGG19, and EfficientNetB3 for accurate and robust classification of rice leaf diseases, using Transfer Learning. ResNet50 achieved maximum accuracy and all other metrics in this comparision.

D. Result of Deep Feature Extraction Method

The Table III depicts the results obtained from 10 runs of training and testing each combination of deep features and classifiers. The results can be summarised as follows:

a. **VGG16 with SVM Classifier:** The extracted features of VGG16 when trained and tested on SVM Classifier achieved perfect result of 1.0 for all metrics. This indicates robust generalization capabilities of this combination.
b. **VGG19 with SVM Classifier** also performed well with accuracy, F1 score, recall and AUC in range 0.998 to 1.0. Highest precision was seen on using Random Forest Classifier. This result demonstrated balanced performance across metrics.
c. **ResNet50 with KNN Classifier** demonstrated almost perfect performance across all metrics (0.999), indicating strong performance in both training and testing phases. It also had the best precision and AUC scores when using Random Forest Classifier, SVM and XGBoost

TABLE III: Results of Classification using Deep Feature Extraction Method

Classifiers	Measures	VGG16	VGG19	RESNET50	EFFICIENTNETB3	INCEPTIONV3
Random Forest Classifier	**Accuracy**	0.960 ± 0.005	0.961 ± 0.005	0.969 ± 0.005	0.974 ± 0.004	0.867 ± 0.007
	Recall	0.960 ± 0.005	0.961 ± 0.005	0.969 ± 0.005	0.974 ± 0.004	0.867 ± 0.007
	Precision	0.999 ± 0.001	**0.999 ± 0.001**	**0.999 ± 0.002**	**0.999 ± 0.001**	**0.979 ± 0.003**
	F1 Score	0.979 ± 0.003	0.980 ± 0.003	0.984 ± 0.002	0.986 ± 0.002	0.912 ± 0.005
	AUC	0.980 ± 0.002	0.981 ± 0.003	0.984 ± 0.002	0.987 ± 0.002	0.930 ± 0.004
Decision Tree Classifier	**Accuracy**	0.910 ± 0.005	0.911 ± 0.005	0.906 ± 0.009	0.907 ± 0.005	0.829 ± 0.007
	Recall	0.910 ± 0.005	0.911 ± 0.005	0.906 ± 0.009	0.907 ± 0.005	0.829 ± 0.007
	Precision	0.912 ± 0.005	0.913 ± 0.005	0.906 ± 0.009	0.909 ± 0.005	0.828 ± 0.007
	F1 Score	0.909 ± 0.005	0.911 ± 0.005	0.905 ± 0.010	0.906 ± 0.006	0.827 ± 0.008
	AUC	0.940 ± 0.003	0.941 ± 0.003	0.937 ± 0.006	0.938 ± 0.004	0.886 ± 0.005
KNN Classifier (N = 5)	**Accuracy**	0.980 ± 0.000	0.981 ± 0.000	**0.999 ± 0.000**	**0.998 ± 0.000**	0.829 ± 0.000
	Recall	0.980 ± 0.000	0.981 ± 0.000	**0.999 ± 0.000**	**0.998 ± 0.000**	0.829 ± 0.000
	Precision	0.980 ± 0.000	0.982 ± 0.000	**0.999 ± 0.000**	0.998 ± 0.000	0.854 ± 0.000
	F1 Score	0.980 ± 0.000	0.981 ± 0.000	**0.999 ± 0.000**	**0.998 ± 0.000**	0.837 ± 0.000
	AUC	0.987 ± 0.000	0.988 ± 0.000	0.999 ± 0.000	0.998 ± 0.000	0.891 ± 0.000
SVM Classifier	**Accuracy**	**1.000 ± 0.000**	**0.998 ± 0.000**	0.998 ± 0.000	**0.998 ± 0.000**	0.814 ± 0.000
	Recall	**1.000 ± 0.000**	**0.998 ± 0.000**	0.998 ± 0.000	**0.998 ± 0.000**	0.814 ± 0.000
	Precision	**1.000 ± 0.000**	0.998 ± 0.000	0.998 ± 0.000	0.998 ± 0.000	0.818 ± 0.000
	F1 Score	**1.000 ± 0.000**	**0.998 ± 0.000**	0.998 ± 0.000	**0.998 ± 0.000**	0.807 ± 0.000
	AUC	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**	0.951 ± 0.000
XGBoost Classifier	**Accuracy**	0.996 ± 0.000	0.996 ± 0.000	0.998 ± 0.000	0.995 ± 0.000	**0.950 ± 0.000**
	Recall	0.996 ± 0.000	0.996 ± 0.000	0.998 ± 0.000	0.995 ± 0.000	**0.950 ± 0.000**
	Precision	0.996 ± 0.000	0.996 ± 0.000	0.998 ± 0.000	0.995 ± 0.000	0.952 ± 0.000
	F1 Score	0.996 ± 0.000	0.996 ± 0.000	0.998 ± 0.000	0.995 ± 0.000	**0.949 ± 0.000**
	AUC	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**	**0.994 ± 0.000**

Highlited values show the best metrics of respective CNN model

Classifiers. This shows that the deep features extracted from ResNet50 were successful in identifying almost all images.

d. **EfficientNetB3** exhibited similarly high performance with KNN and SVM classifier where all metrics were in range of 0.998 to 1.0, it was trailed by XGBoost Classifier. This result depict consistent performance of this model across various metrics.
e. **InceptionV3** gave it's best performance on using XGBoost classifier with all metrics in range of 0.94 to 0.99. It did not perform well with other classifiers, making it the least performing model out of all others.

Overall, the models VGG16, VGG19, ResNet50 and EfficientNetB3 demonstrated robust performance in classifying rice leaf diseases. The combination of these deep learning models and various classifiers (RandomForest, SVM, DecisionForest, KNN and XGBoost) yielded high accuracies, often nearing or achieving perfect scores. The integration of feature extraction with shallow machine learning classifiers gave optimal results, indicating strong generalization and balanced performance across different data splits.

E. Result of Deep Feature Fusion Method

The Table IV shows performances of concatenated extracted deep features trained and tested on various shallow machine classifiers. The results can be summarised as follows:

a. **VGG16 + VGG19 with SVM Classifier:** The concatenated features of VGG16 & VGG19 when trained and tested on SVM classifier gave a perfect result of 1.0 for all metrics. This indicates that extracted deep features of both the models complement each other perfectly and successfully identify the rice leaf diseases.
b. **VGG19 + ResNet50** with SVM and XGBoost classifiers gave almost perfect results where all metrics were in

TABLE IV: Performance of Deep Feature Fusion with Multiple Classifiers

Classifiers	Measures	VGG16 + VGG19	VGG19 + ResNet50	ResNet50 + EfficientNetB3	ResNet50 + InceptionV3	EfficientNetB3 + InceptionV3
Random Forest Classifier	**Accuracy**	0.967 ± 0.006	0.976 ± 0.004	0.977 ± 0.004	0.963 ± 0.005	0.962 ± 0.004
	Recall	0.967 ± 0.006	0.976 ± 0.004	0.977 ± 0.004	0.963 ± 0.005	0.962 ± 0.004
	Precision	0.999 ± 0.001	0.998 ± 0.001	0.999 ± 0.001	**0.998 ± 0.001**	0.999 ± 0.001
	F1 Score	0.983 ± 0.003	0.987 ± 0.002	0.988 ± 0.002	0.980 ± 0.002	0.980 ± 0.003
	AUC	0.983 ± 0.003	0.988 ± 0.002	0.988 ± 0.002	0.981 ± 0.002	0.981 ± 0.002
Decision Tree Classifier	**Accuracy**	0.923 ± 0.003	0.932 ± 0.004	0.916 ± 0.005	0.912 ± 0.006	0.926 ± 0.007
	Recall	0.923 ± 0.003	0.932 ± 0.004	0.916 ± 0.005	0.912 ± 0.006	0.926 ± 0.007
	Precision	0.924 ± 0.003	0.933 ± 0.004	0.916 ± 0.006	0.912 ± 0.006	0.927 ± 0.007
	F1 Score	0.923 ± 0.003	0.932 ± 0.004	0.916 ± 0.006	0.911 ± 0.006	0.926 ± 0.007
	AUC	0.949 ± 0.002	0.955 ± 0.003	0.944 ± 0.004	0.941 ± 0.004	0.951 ± 0.005
KNN Classifier (N= 5-10)	**Accuracy**	0.976 ± 0.011	0.988 ± 0.007	0.989 ± 0.006	0.776 ± 0.030	0.768 ± 0.034
	Recall	0.976 ± 0.011	0.988 ± 0.007	0.989 ± 0.006	0.776 ± 0.030	0.768 ± 0.034
	Precision	0.981 ± 0.007	0.993 ± 0.005	0.993 ± 0.004	0.847 ± 0.018	0.841 ± 0.021
	F1 Score	0.979 ± 0.009	0.990 ± 0.005	0.991 ± 0.004	0.799 ± 0.021	0.792 ± 0.025
	AUC	0.985 ± 0.006	0.993 ± 0.004	0.993 ± 0.003	0.866 ± 0.015	0.861 ± 0.017
SVM Classifier	**Accuracy**	**1.000 ± 0.000**	**0.998 ± 0.000**	0.998 ± 0.000	0.838 ± 0.000	0.816 ± 0.000
	Recall	**1.000 ± 0.000**	**0.998 ± 0.000**	0.998 ± 0.000	0.838 ± 0.000	0.816 ± 0.000
	Precision	**1.000 ± 0.000**	**0.998 ± 0.000**	0.998 ± 0.000	0.841 ± 0.000	0.820 ± 0.000
	F1 Score	**1.000 ± 0.000**	**0.998 ± 0.000**	0.998 ± 0.000	0.833 ± 0.000	0.810 ± 0.000
	AUC	**1.000 ± 0.000**	**0.998 ± 0.000**	**1.000 ± 0.000**	0.963 ± 0.000	0.952 ± 0.000
XGBoost Classifier	**Accuracy**	0.994 ± 0.000	0.996 ± 0.000	**1.000 ± 0.000**	**0.998 ± 0.000**	**1.000 ± 0.000**
	Recall	0.994 ± 0.000	0.996 ± 0.000	**1.000 ± 0.000**	**0.998 ± 0.000**	**1.000 ± 0.000**
	Precision	0.994 ± 0.000	0.996 ± 0.000	**1.000 ± 0.000**	**0.998 ± 0.000**	**1.000 ± 0.000**
	F1 Score	0.994 ± 0.000	0.996 ± 0.000	**1.000 ± 0.000**	**0.998 ± 0.000**	**1.000 ± 0.000**
	AUC	**1.000 ± 0.000**	**0.998 ± 0.000**	**1.000 ± 0.000**	**0.998 ± 0.000**	**1.000 ± 0.000**

Highlighted values show the best metrics of respective feature fusion

range of 0.996 to 0.999. Giving high accuracy over multiple classifiers indicates strong generalization capability of this combination.

c. **ResNet50 + EfficientNetB3 with XGBoost** reached perfect result of 1.0 for all metrics, and on SVM and KNN the metrics were in range 0.98 to 1.0. This result depicts constant success over multiple classifiers.

d. **ResNet50 + InceptionV3** combination gave good result of around 0.99 for all metrics on XGBoost Classifier, but it's results were less than 0.9 on all metrics for other classifiers. This shows that this combination of features is useful only with XGBoost classifier.

d. **EfficientNetB3 + InceptionV3 with XGBoost Classifier** concatenated deep features when trained and tested on XGBoost classifier gave perfect result of 1.0 for all metrics. InceptionV3 did not perform well in transfer learning but when it's features are combined with EfficientNetB3's deep features, it gives perfect result for XGBoost Classifier.

Overall, this method yielded better result than transfer learning, as it takes less time to process and it also gives better metrics when compared to individual feature extraction and classification process.

F. Result of Handcrafted Features and Fusion with Deep Features

At first we extracted various handcrafted features from the dataset : HOG, LBP, GLCM, SIFT and Gabor Features. Then we trained these features and tested them on various classifiers. The accuracy obtained using these features is compared in Table V.

Results show that GLCM features when trained and tested on XGBoost classifier gave the highest accuracy of 0.99. But GLCM did not perform very well with SVM and KNN classifiers where it's accuracy was in range 0.53 to 0.87. When observed closely the best overall performance was obtained using LBP features, whose accuray was in range 0.95 to 1.0 for all classifiers except SVM. For further experiments, we decided to use LBP features because of it's capability for generalization and fast extraction process.

In the next steps we concatenated the features from LBP with deep features of 5 pre trained CNN models, one at a time, and then trained and tested the concatenated features on various machine learning classifiers. The results are shown in table VI, and can be summarised as follows:

a. **VGG16 + LBP:** The concatenation of deep features of VGG16 and LBP Features, when trained and tested on SVM Classifier gave a perfect result of 1.0 for all metrics. his indicates that extracted deep features and

TABLE V: Accuracy Comparison of Handcrafted Features on Multiple Classifiers

Classifiers	HOG	LBP	GLCM	Gabor Features	SIFT
Random Forest Classifier	0.430 ± 0.007	**0.998 ± 0.001**	0.990 ± 0.002	0.994 ± 0.001	0.399 ± 0.008
Decision Tree Classifier	0.586 ± 0.010	0.949 ± 0.005	**0.978 ± 0.005**	0.953 ± 0.005	0.559 ± 0.008
KNN Classifier (N = 5)	0.638 ± 0.000	0.978 ± 0.000	0.869 ± 0.000	**0.980 ± 0.000**	0.411 ± 0.000
SVM Classifier	**0.891 ± 0.000**	0.645 ± 0.000	0.526 ± 0.000	0.588 ± 0.000	0.511 ± 0.000
XGBoost Classifier	0.866 ± 0.000	0.998 ± 0.000	**0.999 ± 0.000**	0.998 ± 0.000	0.648 ± 0.000

Highlighted values show the highest accuracy

TABLE VI: Accuracy Comparison of Handcrafted Features on Multiple Classifiers

Classifiers	Measures	VGG16 + LBP	VGG19 + LBP	ResNet50 + LBP	EfficientNetB3 + LBP	InceptionV3 + LBP
Random Forest Classifier	**Accuracy**	0.967 ± 0.007	0.970 ± 0.007	0.973 ± 0.002	0.974 ± 0.004	0.883 ± 0.006
	Recall	0.967 ± 0.007	0.970 ± 0.007	0.973 ± 0.002	0.974 ± 0.004	0.883 ± 0.006
	Precision	0.999 ± 0.001	**0.999 ± 0.001**	0.997 ± 0.002	0.999 ± 0.001	**0.979 ± 0.003**
	F1 Score	0.982 ± 0.004	0.984 ± 0.004	0.985 ± 0.002	0.986 ± 0.002	0.922 ± 0.003
	AUC	0.983 ± 0.004	0.985 ± 0.004	0.986 ± 0.001	0.987 ± 0.002	0.938 ± 0.003
Decision Tree Classifier	**Accuracy**	0.923 ± 0.008	0.926 ± 0.008	0.913 ± 0.006	0.906 ± 0.006	0.893 ± 0.008
	Recall	0.923 ± 0.008	0.926 ± 0.008	0.913 ± 0.006	0.906 ± 0.006	0.893 ± 0.008
	Precision	0.923 ± 0.007	0.928 ± 0.009	0.914 ± 0.006	0.909 ± 0.006	0.894 ± 0.008
	F1 Score	0.922 ± 0.008	0.926 ± 0.008	0.913 ± 0.006	0.904 ± 0.006	0.893 ± 0.008
	AUC	0.948 ± 0.005	0.950 ± 0.005	0.942 ± 0.004	0.937 ± 0.004	0.929 ± 0.005
KNN Classifier	**Accuracy**	0.961 ± 0.013	0.972 ± 0.009	0.988 ± 0.005	0.989 ± 0.005	0.767 ± 0.033
	Recall	0.961 ± 0.013	0.972 ± 0.009	0.988 ± 0.005	0.989 ± 0.005	0.767 ± 0.033
	Precision	0.969 ± 0.008	0.981 ± 0.004	0.993 ± 0.004	0.993 ± 0.003	0.840 ± 0.020
	F1 Score	0.964 ± 0.010	0.976 ± 0.006	0.990 ± 0.004	0.991 ± 0.004	0.791 ± 0.024
	AUC	0.975 ± 0.007	0.983 ± 0.005	0.993 ± 0.003	0.993 ± 0.003	0.860 ± 0.017
SVM Classifier	**Accuracy**	**1.000 ± 0.000**	**0.998 ± 0.000**	**0.998 ± 0.000**	0.998 ± 0.000	0.814 ± 0.000
	Recall	**1.000 ± 0.000**	**0.998 ± 0.000**	**0.998 ± 0.000**	0.998 ± 0.000	0.814 ± 0.000
	Precision	**1.000 ± 0.000**	0.998 ± 0.000	**0.998 ± 0.000**	0.998 ± 0.000	0.818 ± 0.000
	F1 Score	**1.000 ± 0.000**	**0.998 ± 0.000**	**0.998 ± 0.000**	0.998 ± 0.000	0.807 ± 0.000
	AUC	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**	0.951 ± 0.000
XGBoost Classifier	**Accuracy**	0.989 ± 0.000	0.994 ± 0.000	0.995 ± 0.000	**1.000 ± 0.000**	**0.975 ± 0.000**
	Recall	0.989 ± 0.000	0.994 ± 0.000	0.995 ± 0.000	**1.000 ± 0.000**	**0.975 ± 0.000**
	Precision	0.989 ± 0.000	0.994 ± 0.000	0.995 ± 0.000	**1.000 ± 0.000**	0.976 ± 0.000
	F1 Score	0.989 ± 0.000	0.994 ± 0.000	0.995 ± 0.000	**1.000 ± 0.000**	**0.975 ± 0.000**
	AUC	0.999 ± 0.000	0.999 ± 0.000	1.000 ± 0.000	**1.000 ± 0.000**	**0.999 ± 0.000**

Highlighted values show the best metrics for respective model combination

LBP features were able to capture all important features and successfully identified the rice leaf diseases.

b. **EfficientNetB3 + LBP:** The concatenation of deep features of EfficientNetB3 and LBP Features, when trained and tested on SVM Classifier gave a perfect result of 1.0 for all metrics. his indicates that extracted deep features and LBP features were able to capture all important features and successfully identified the rice leaf diseases.

c. **VGG19 + LBP and ResNet50 + LBP:** The concatenation of these two model's deep features and LBP Features, when trained and tested on SVM Classifier gave almost perfect result in range 0.998 to 1.0 for all metrics. Giving

TABLE VII: Performance of VGG19 + ResNet50 + LBP feaures across multiple classifiers

Classifiers	Accuracy	Recall	Precision	F1 Score	AUC
Random Forest Classifier	0.979 ± 0.005	0.979 ± 0.005	**0.999 ± 0.002**	0.989 ± 0.003	0.989 ± 0.003
Decision Tree Classifier	0.925 ± 0.007	0.925 ± 0.007	0.925 ± 0.007	0.925 ± 0.007	0.950 ± 0.004
KNN Classifier (N = 5)	0.988 ± 0.007	0.988 ± 0.007	0.993 ± 0.005	0.990 ± 0.005	0.993 ± 0.004
SVM Classifier	**0.998 ± 0.000**	**0.998 ± 0.000**	0.998 ± 0.000	**0.998 ± 0.000**	**1.000 ± 0.000**
XGBoost Classifier	0.996 ± 0.000	0.996 ± 0.000	0.996 ± 0.000	0.996 ± 0.000	**1.000 ± 0.000**

Highlighted values show the highest values of each metric

high accuracy over multiple classifiers indicates strong generalization capability of this combination.

d. **InceptionV3 + LBP:** The concatenation of deep features of InceptionV3 and LBP features when trained and tested on XGBoost classifier gave a good performance in range 0.97 to 0.99 across all metrics, but did not perform well with other classifiers. This shows that this combination is suitable to use with XGBoost classifier.

Overall, we can safely say that concatenation of deep features of pre trained CNN models and handcrafted features like LBP can provide better results than transfer learing and feature extraction methods which were tried before.

To observe what happens when we concatenate more than two CNN model's deep features with handcrafted features, we tried to concatenate LBP features with extracted deep features of VGG19 and ResNet50. The result of fusion of all threee features are shown in table VII.

e. **VGG19 + ResNet50 + LBP:** The fusion of deep features of VGG19 and ResNet50 had performed very well in feature fusion method ranging from 0.93 to 0.98 for all metrics across all classifiers. When concatenated with LBP features all metrics performed almost perfectly with score of 0.998 to 1.0 with SVM Classifier. This shows that concatenating more than two deep features with handcrafted feature improve the performance of a model.

G. Discussion

In evaluating the performance of various models, we explored whether any techniques yield nearly perfect results in predicting the correct classification. The answer is affirmative; several techniques demonstrated near-perfect or perfect performances. Transfer learning on ResNet50 showed perfect result. Models combining VGG16, VGG19, and ResNet50 with classifiers such as SVM and KNN frequently achieved perfect scores across all metrics, including accuracy, precision, recall, F1 score, and specificity. This indicates these combinations have strong generalization capabilities and balanced performance across different data splits. Combining multiple feature extraction techniques, such as combining LBP with deep learning models significantly enhanced model performance, often resulting in perfect or near-perfect accuracies and other metrics. This demonstrates that integrating various feature extraction methods can provide comprehensive and robust feature representations, thereby improving classification accuracy. However, certain limitations were observed with specific models; for example, the combination of LBP features with InceptionV3 showed moderate performance with lower test accuracy and F1 scores compared to other methods. While these traditional feature extraction methods are useful, they may not be as effective as deep learning models for complex image classification tasks. These findings align with existing literature that emphasizes the efficacy of deep learning models and advanced machine learning techniques in automated disease diagnosis. The superior performance of combination of deep features of CNN models like VGG16, VGG19, ResNet50 and EfficientNetB3 with handcrafted features like LBP, trained and tested on SVM or XGBoost classifiers, corroborates previous research that highlights the potential of these methods in achieving high classification accuracy and robust generalization, further extending their successful application to rice leaf disease classification.

CONCLUSION

In conclusion, this study explored the effectiveness of various deep learning models and traditional feature extraction techniques and their various combinations in classifying rice leaf diseases. The results underscored the superior performance of combination of deep features of CNN models like VGG16, VGG19, ResNet50 and EfficientNetB3 with handcrafted features like LBP, trained and tested on SVM or XGBoost classifiers, achieving near-perfect classification accuracy across multiple metrics. The combination of advanced deep learning architectures with robust feature extraction methods yielded significant improvements in classification outcomes, demonstrating their potential for automated disease diagnosis in agriculture.

Moving forward, future research should focus on several key areas. Firstly, exploring ensemble methods that combine multiple models could further enhance classification robustness and accuracy. Additionally, investigating transfer learning techniques across different crop types beyond rice could broaden the applicability of these models in agricultural settings.

Furthermore, incorporating real-time disease monitoring and prediction capabilities using IoT and remote sensing technologies would facilitate proactive disease management strategies. Finally, addressing the scalability and computational efficiency of these models for deployment in resource-constrained environments remains crucial for practical implementation.

By addressing these avenues, future studies can build upon the current findings to develop more effective and scalable solutions for automated disease diagnosis in agriculture, thereby contributing to sustainable crop management and food security initiatives globally.

[1] [2] [3] [4] [5]

REFERENCES

[1] Prabira Kumar Sethy, Nalini Kanta Barpanda, Amiya Kumar Rath, and Santi Kumari Behera. Deep feature based rice leaf disease identification using support vector machine. *Computers and Electronics in Agriculture*, 175:105527, 2020.

[2] Xijian Fan, Peng Luo, Yuen Mu, Rui Zhou, Tardi Tjahjadi, and Yi Ren. Leaf image based plant disease identification using transfer learning and feature fusion. *Computers and Electronics in agriculture*, 196:106892, 2022.

[3] Muhammad Sharif, Muhammad Attique Khan, Zahid Iqbal, Muhammad Faisal Azam, M Ikram Ullah Lali, and Muhammad Younus Javed. Detection and classification of citrus diseases in agriculture based on optimized weighted segmentation and feature selection. *Computers and Electronics in Agriculture*, 150:220–234, 2018.

[4] Thomas Serre, Lior Wolf, and Tomaso Poggio. Object recognition with features inspired by visual cortex. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 994–1000. IEEE, 2005.

[5] Hafiz Tayyab Rauf, M Ikram Ullah Lali, Saliha Zahoor, Syed Zakir Hussain Shah, Abd Ur Rehman, and Syed Ahmad Chan Bukhari. Visual features based automated identification of fish species using deep convolutional neural networks. *Computers and Electronics in Agriculture*, 167:105075, 2019.