University of Hertfordshire

School of Physics, Engineering and Computer Science

Advanced Computer Science Masters Project

7COM1039-0901-2024

Project Title:

Enhancing Rice Crop Health Monitoring: CNN-Based Detection of Rice Leaf Diseases

By

Danish Jalil

Student ID

22016088

Supervisor:	*Dr. Muawya eldaw*
Date Submitted:	*01/06/2025*
Word Count:	*9136*
GitHub address:	*Click here to access GitHub*

DECLARATION STATEMENT

This report is submitted in partial fulfilment of the requirement for the degree of Master of Science in Data Science at the University of Hertfordshire. I have read the guidance to students on academic integrity, misconduct and plagiarism information at Assessment Offences and Academic Misconduct and understand the University process of dealing with suspected cases of academic misconduct and the possible penalties, which could include failing the project module or course. I certify that the work submitted is my own and that any material derived or quoted from published or unpublished work of other persons has been duly acknowledged. (Ref. UPR AS/C/6.1, section 7 and UPR AS/C/5, section 3.6). I have not used chatGPT, or any other generative AI tool, to write the report or code (other than where declared or referenced). I did not use human participants or undertake a survey in my MSc Project.

I hereby give permission for the report to be made available on module websites provided the source is acknowledged.

Student SRN number: 22016088

Student Name printed: Danish Jalil

Student signature: Danish Jalil

DEDICATION

It is with great pleasure that I dedicate this research to my family, my colleagues, and my friends, who have continually helped me throughout the whole process of this research with encouragement and unwavering belief. Their support was especially valuable during the times when I felt discouraged and challenged in my work.

ACKNOWLEDGMENTS

It is my sincere pleasure to express my sincere gratitude to my thesis supervisor, Dr. Muawya eldaw, for providing me with steadfast guidance as well as inspiring mentoring throughout the course of my research. In addition, I must acknowledge the incredible support of my family, whose encouragement and motivation have been crucial to bringing this research to fruition. His enthusiasm, patience, and insightful feedback have greatly influenced my work.

Abstract

The rice leaf images were classified using convolutional neural networks (CNNs) and MobileNets based on factors such as bacterial leaf blight, brown spots, and leaf smut. As a result of the classification of 120 high-resolution images into three different types of diseases, we were able to identify the three diseases: Bacterial Leaf Blight, Brown Spots, and Leaf Smut. On the training data, both models achieved 100% accuracy, with CNN exhibiting a significantly lower training loss, suggesting overfitting to the model. As compared to CNN, MobileNet achieved better generalization on the validation set, achieving 91.67% accuracy. While the CNN model excelled at detecting Bacterial Leaf Blight, it struggled to detect Brown Spot and Leaf Smut, exhibiting lower recall and F1-scores for these diseases. For Brown Spot and Leaf Smut, MobileNet showed balanced and higher performance across all disease categories. The results of this comparative analysis demonstrate MobileNet's superior ability to generalize to unseen data and its potential for real-world applications in automating rice disease diagnosis. Agricultural automation, specifically the detection of rice diseases, may benefit from the advancement of deep learning techniques, such as MobileNet.

Keyword : Agricultural automation, image processing, deep learning, rice disease detection, MobileNet, CNN

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. CHAPTER I: INTRODUCTION

1.1. Problem Overview

Rice is a crucial source of nutrition globally, supporting food security and the economies of many nations(Bandumula 2018). However, agricultural challenges, particularly diseases affecting rice plants, pose significant risks to productivity. In countries like Bangladesh and India, where the majority of the population is dependent on agriculture, rice diseases such as Brown Spot, Tungro, Sheath Blight, Bacterial Leaf Blight, and Bacterial Leaf Blast are major concerns (N. Kumar et al. 2021; Mukherjee et al. 2025; Zhao, Lin, and H. Chen 2020). The prevalence of these diseases threatens crop yield, quality, and food security, especially in rural areas where farmers struggle to identify and manage these issues.The difficulty in diagnosing rice leaf diseases is compounded by the overlapping symptoms of various diseases, often leading to confusion and ineffective treatments(Strange and Scott 2005). Moreover, climatic factors exacerbate the situation, making timely identification and intervention even more difficult. The need for a precise and reliable diagnostic method for early disease detection in rice crops has become paramount. Agricultural performs can be transformed through AI-driven tools that deliver accurate and effectual disease classification and running, which can convert agricultural performs in this context (Organization et al. 2013).

1.2. Current Issues

The rice segment faces several dangerous challenges connected to crop illnesses, which meaningfully delay yield and excellence. In republics like Bangladesh and India, where rice is important for both food and financial stability, diseases such as bacterial blight, fungal infections, and other leaf diseases lead to substantial losses(Fahad et al. 2019). Early detection and exact analysis are vital to extenuating the blowout of these illnesses and minimalizing the financial impact. Insufficient disease documentation methods, such as graphic reviews by farmers, are time-consuming and often imprecise. This is mainly problematic for large-scale rice agricultural, where physical monitoring is not feasible(Buja et al. 2021). Furthermore, the impact of climate change and increased ecological pressures have made the rice crop more vulnerable to illnesses. Without quick and precise diagnosis, disease occurrences can consequence in abridged crop yields, monetary loss, and food security pressures.

1.3. Project Details

Rice leaf illnesses require deep learning models, such as Convolutional Neural Networks (CNNs) and MobileNet. This study will evaluate CNNs based on images of Brown Spot, Leaf Spot, and Bacterial Blight on rice plants on Kaggle(Hasan et al. 2023). The perfect will contribution in early detection and organisation of disease by sleuthing brown spots, black dots, and water-soaked cuts. As deep learning techniques have increased purchase in agriculture, they are able to deliver faster, more accurate, and automated identifies, which decreases the need for expert information (Wani et al. 2022). For classifying complex designs in rice leaf imageries, CNNs excel at image classification errands.

1.4. Aims and Objectives

Using CNNs and mobilnet for initial disease detection and organisation is the aim of this study. As a consequence of this study, the next objectives will be attained:

1. To provide an impression of the trials associated with rice leaf disease detection and its significance for agriculture and food safety.
2. To design and implement a CNNs-based model for identifying and classifying rice diseases based on Rice leaf images.
3. To review and evaluate current methods of plant disease identification using deep learning methods.
4. To analyze the performance of the CNN model trained on a Kaggle rice leaf disease dataset, comparison it to traditional investigative methods.
5. To assess the potential benefits of AI-driven solutions in improving disease management and mitigating crop losses.

1.5. Research Questions

It is the aim of this study to address the following research questions:

1. Can deep learning models accurately detect sick leaves in rice plants?
2. How can deep learning-based disease detection aid in the diagnosis of rice leaf diseases in agricultural practices?
3. What is the potential impact of this technology on the agricultural sector and the broader economy?

1.6. Motivation of This Study

Farmers have long relied on visual inspection and expert knowledge to identify and manage plant diseases, a time-consuming, error-prone, and inefficient procedure. Plant diseases are increasing due to climate change and environmental stressors, complicating traditional diagnostic methods (John et al. 2023).It is consequently imperative that farmers are helped in detecting and treating crop diseases in a timely manner. Deep learning and computer vision can help resolve these challenges (Amulothu et al. n.d.). Models using deep learning can process large amounts of data and detect disease suggestions accurately, reducing Labor and costs. By means of a rapid, reliable, and available disease detection system, this study aims to improve agricultural does, improve crop health, and decrease financial losses (Bao et al. 2019).

1.7. Novelty of the Study

In this study, progressive deep learning methods, precisely CNNs, are applied to Kaggle datasets of rice leaf images to detect rice leaf disease. Founded on visual symptoms, the model can accurately categorise various rice diseases (Upadhyay and A. Kumar 2022). This method offers farmers a scalable, automatic, and real-time solution that decreases crop losses due to undiagnosed illnesses (Jafar et al. 2024). Furthermore, this investigate donates to the growing body of information on AI applications in agriculture, chiefly in rice farming (Ahmad et al. 2024).

1.8. Final Feasibility, Commercial Context, and Risk

The possibility of this profound learning-based illness detection system is talented, as CNNs have confirmed strong presentation in alike image organisation tasks(Razzak, Naz, and Zaib 2018). The model, once industrialised and tested, can be organized through mobile apps or other stages that allow farmers to take images of their crops and obtain instant disease identifies. This could significantly decrease the time and capitals spent on disease organisation and improve the overall output of rice farming(Singh 2018).In footings of commercialization, this knowledge has the potential to be climbed across various agricultural sectors, counting in republics where rice is a major crop. It could be advertised as part of a exactness agriculture platform, contribution not just disease detection but also references for treatment, pest control, and crop management(Awotide, Karimov, and Diagne 2016). The model, however, expressions risks such as its accuracy under dissimilar environmental circumstances, the need to train on a great and diverse dataset, and its early cost. Ensuring that the model is flexible to numerous climatic conditions and user-friendly for farmers with incomplete technical knowhow will be indispensable for its success in the field.

1.9. Commercial Context

Given that over 80% of the population in countries like India and Bangladesh depend on agriculture, the introduction of AI-driven solutions for disease detection could lead to significant improvements in agricultural productivity and food security(Dhal and Kar 2024; Mohajan 2013). The ability to detect and manage rice leaf diseases quickly can help prevent widespread crop losses and ensure a steady food supply(Balasubramanian et al. 2007).The commercial potential for such AI-based solutions is substantial, not only in rice farming but also in other crops and regions, as the technology could be adapted for various agricultural contexts worldwide.

1.10. Report structure

This report is structured into the following key sections. The **Literature** section reviews existing research, identifying gaps and challenges that the study aims to address. The **Methodology** outlines the approach, techniques, and tools used, providing a detailed explanation of data collection, analysis, and any models or frameworks applied. The **Experimental Results** present the findings of the study, supported by tables, graphs, or figures, and discuss observed patterns and their implications. Finally, the **Conclusion** summarizes the key outcomes, highlights contributions, acknowledges limitations, and suggests directions for future research. This structure ensures a logical flow from existing knowledge to new insights.

2. CHAPTER II: LITERATURE REVIEW

2.1. RELATED WORK

2.2. Introduction to the Field

There are many diseases that affect the color, size, or form of rice around the world, which reduces its productivity. It is estimated that rice diseases contribute to a 37 percent decrease in rice production decline in rice productivity every year, according to Dhaka et al. (Dhaka et al. 2021). These diseases are not only difficult to diagnose, but their detection and management are also poorly understood. Machine learning and deep learning have made it possible to identify and manage rice diseases efficiently. The aim of this section is to review research on deep learning models applied to rice plant disease detection and diagnosis. Even though an appropriate method for detecting and controlling rice illnesses has not yet been developed, there is still a lack of knowledge about the identification and management of these illnesses. Despite this, very few studies have been conducted on diagnosing rice plant diseases. It also discusses new research on deep learning models for rice disease diagnosis (Sethy et al. 2020b).

2.3. Key Studies and Works

The following research highlights various approaches and models used for detecting rice plant diseases:

The Faster R-CNN algorithm was used by Bari et al. (Bari et al. 2021) to distinguish among rice illnesses including blast, brown spot, and Hispa. They were able to determine leaves that were healthy with an accuracy of 99.25%. Both a self-generated dataset and a Kaggle dataset with 2,400 photos in total were used to assess the models. In addition, characteristic maps of diseased leaves were employed for training in a Caffe deep learning technique. An internet database and their own dataset were used to generate a unique Rice Leaf Disease Dataset (RLDD).

Pandian et al.(Pandian J et al. 2022) ResNet19 layers were optimized using evolutionary search. We used data augmentation techniques such as growing, trimming, tumbling, padding, and rotation to add more data to pre-existing photographs. A ResNet197 model was trained using 154,500 photos from 22 different plants, including both normal and sick leaves. The input images were 224 by 224 × 3 pixels. The model achieved 99.58% classification accuracy after training for up to 1,000 epochs in a GPU environment.

A framework for prompt identification of illnesses was created by Roy and Bhaduri(Roy and Bhaduri 2021), who aimed to maximise the rapidnessand precision of identifying apple illnesses in intricate orchard settings. Their simulation was processed at 56.9 frames per second

(FPS) and had a mean average precision (mAP) of 91.2% and an F1 score of 95.9%. With a mAP gain of 9% and an F1 score improvement of 7.6%, this method demonstrated notable improvements over earlier models.

Saberi Anari (Saberi Anari 2022) diagnosed damaged leaves with Model Engineering. Using deep transfer learning, this method combines radial basis functions, k-NN, decision trees, neural networks, and ensembles to obtain characteristics of images. Specifically, deep transfer learning was used to extract characteristics from pictures of leaves of different fruits.

In order to train and evaluate 5 separate models, Deng et al. (Deng et al. 2021) gathered 33,026 photos of six distinct rice illnesses of the leaves. The methodologies, dataset, camera specs, amount of findings, training rate, frequency of rounds, and model efficacy were all thoroughly described in the paper.

To diagnose rice leaf illnesses, Zhou et al. (Zhou et al. 2023) used Faster R-CNN fusion in conjunction with FCM-KM clustering. 3,010 photos made up the dataset, which had 15,000 rounds and a learning rate of 0.001. For rice explosion, microbial decay, and pollution, the model's accuracy was 96.71%, 97.53%, and 98.26%, respectively. Sethy et al.(Sethy et al. 2020a) used a dataset collected with a mobile phone camera in a farm area for identifying rice fake smuts utilising a Faster R-CNN model. The effectiveness of the model was not disclosed, however the dataset had 50 assessments, 5 rounds, and a learning rate of 0.001.

An improved method was used by Ramesh and Vydeki (Ramesh and Vydeki 2020) to diagnose rice leaf illnesses, and the dataset was also taken from an agricultural area. Rice explosion, microbiological destruction, sheathing rot, and brown spot all had rates of success of 98.9%, 95.78%, and 94%, respectively, for the model they used. A Faster R-CNN model was employed by Deng et al. (2021) and Li et al. (2017) to identify rice illnesses of the leaves using a dataset gathered from crops of rice in China's Anhui and Hunan regions. Both a Sony DSC-QX10 camera and a cell phone camera were used to record the data. 5,320 photos made up the dataset, which had 50,000 rounds and a learning rate of 0.002.

Prajapati et al. (Prajapati, Shah, and Dabhi 2017) Rice leaf disease was diagnosed using NIKON D90 digital SLR camera datasets taken on farmland. The acquisition rate and number of rounds were not specified, but 120 samples were collected. As a result of training and testing, the model gave 93.33 percent, 73.3 percent, 83.80 percent, and 88.57 percent during cross-validation five times. The CNN model was employed by Rahman et al. (2020) for accurate categorisation rice leaf disease diagnosis.

Velusamy et al. (Velusamy et al. 2023) used handmade feature development using segmentation, augmentation, and pre-processing to improve the precision of classification. With this method, the final classification accuracy was 90.63%, an improvement of 3.1%.

In order to lessen dependency on conventional methods for protecting the production of rice, Haridasan et al. (Haridasan, Thomas, and Raj 2023) Machine learning, deep learning, and image processing can be combined to diagnose rice plant diseases automatically. The rice plant's afflicted regions are pinpointed using segmentation methods after picture preprocessing.

Visual traits detect disorders. Rice plant diseases are classified and diagnosed using Support Vector Machine (SVM) and Convolutional Neural Network (CNN) classifiers.

2.4. Relation to Your Research and Hypothesis

The summarized studies provide a foundation for understanding current advancements in rice plant disease diagnosis using machine learning and deep learning models. The high accuracy achieved by models like Faster R-CNN and ResNet19 highlights the potential of leveraging sophisticated algorithms to enhance detection performance. However, the variability in dataset sizes, preprocessing techniques, and performance metrics indicates a need for standardized approaches. Additionally, the reliance on specific datasets and limited exploration of hybrid methods suggests opportunities for further innovation.

2.5. Appropriate Sources and Quality

The studies referenced in this research are sourced from reputable journals, conferences, and datasets, ensuring the credibility and reliability of the information. Each source has been selected based on its relevance to rice plant disease diagnosis and the quality of its methodology and results. High-impact studies, such as those employing advanced models like Faster R-CNN and ResNet19, have been included to highlight the current state-of-the-art approaches in the field. Table 2.1 summarizes the existing literature along with their achievements and limitations.

Table 2.1. Summary of Literature on Rice Plant Disease Diagnosis

Author(s)	**Algorithm/Model Used**	**Dataset Details**	**Techniques**	**Performance**
(Bari et al. 2021)	Faster R-CNN	Self-generated and Kaggle datasets with 2,400 images	Characteristic maps, Caffe deep learning	Accuracy: 99.25%
(Pandian J et al. 2022)	ResNet19 with evolutionary search	Pooled dataset with 154,500 images from 22 plants	Data augmentation (growing, trimming, rotation), pre-trained ResNet19, GPU training	Accuracy: 99.58%
(Roy and Bhaduri 2021)	Deep learning framework	Orchard setting dataset (details unspecified)	High FPS processing (56.9), precision-focused training	mAP: 91.2%, F1: 95.9%

Author(s)	Algorithm/Model Used	Dataset Details	Techniques	Performance
(Saberi Anari 2022)	Deep CNN + SVM models (e.g., RBF, k-NN)	Dataset details unspecified	Deep transfer learning, ensemble SVM models	Performance not specified
(Deng et al. 2021)	Various models	33,026 images of six rice leaf diseases	Thorough methodology description, various camera specs	Performance not specified
(Zhou et al. 2023)	Faster R-CNN + FCM-KM clustering	Dataset of 3,010 images with 15,000 rounds	Hybrid clustering approach	Accuracy: 96.71%-98.26%
(Sethy et al. 2020a)	Faster R-CNN	Mobile camera farm dataset	Minimal rounds (5), learning rate 0.001	Performance not disclosed
(Ramesh and Vydeki 2020)	Improved Faster R-CNN	Agricultural dataset	Comprehensive analysis	Accuracy: 94%-98.9%
(Prajapati, Shah, and Dabhi 2017)	SVM	Nikon D90-based farmland dataset of 120 samples	5/10-fold cross-validation	Accuracy: 93.33%-88.57%
(Velusamy et al. 2023)	CNN-based watershed segmentation	Dataset details unspecified	Handcrafted feature development, segmentation	Accuracy: 90.63%
(Haridasan, Thomas, and Raj 2023)	Automated vision-based approach	Dataset details unspecified	Image preprocessing, segmentation, hybrid SVM-CNN	Performance not specified

2.6. Depth and Breadth of Coverage

The literature reveals that various methodologies, datasets, and technologies have been applied to rice disease diagnosis, showcasing depth and breadth in approaches:

- Studies like that of Zhou et al. (Zhou et al. 2023) incorporated advanced clustering methods (FCM-KM) alongside Faster R-CNN to achieve accuracies exceeding 96%.
- Deng et al. (Deng et al. 2021) explored the use of multiple models trained on a large dataset of 33,026 images, providing a detailed description of the methodologies, camera specifications, and dataset variations.
- The work by Haridasan et al. (Haridasan, Thomas, and Raj 2023) demonstrates the

integration of machine learning and deep learning methods with preprocessing and segmentation to enhance precision in disease detection.

2.7. Comparative Analysis

Comparing the models and methodologies reveals both strengths and limitations:

2.7.1. Strengths

- Models like ResNet19 (Pandian J et al. 2022) and Faster R-CNN (Bari et al. 2021) exhibit high classification accuracy due to robust architectures and comprehensive datasets.
- Techniques such as data augmentation and clustering have significantly enhanced accuracy in disease diagnosis (Zhou et al. 2023).

2.7.2. Weaknesses

- Some studies lack detailed performance metrics or dataset descriptions, which hampers replicability and generalization.

2.8. Identification of Gaps

Despite significant advancements, several gaps remain in the current research:

1. Many studies lack severity-based classification methods, which could aid in better disease management strategies.
2. A comprehensive, standardized dataset for rice disease detection is absent, leading to variations in model performance and generalizability.
3. Few models address real-time deployment in field conditions, as highlighted by the limited FPS in simulations like Roy and Bhaduri's work (Roy and Bhaduri 2021).

3. CHAPTER III: METHODOLOGY

3.0.1. Project Design

This section presents the project design . First, I extracted images from the dataset. After applying pre-processing on the whole dataset, a matrix of features was created using the Red, Green, and Blue intensity standards. The framework was then provided with this characteristic vector so that CNN and Mobile Net models used and trained on the Rice Leaf Diseases datasets. In the end, the accuracy was calculated and analyzed to see how well each model worked on the dataset. The schematic structure that represents the project design used in this work is shown in Fig. 3.1. It is noteworthy that the Rice Leaf illnesses datasets used in this work is openly accessible.

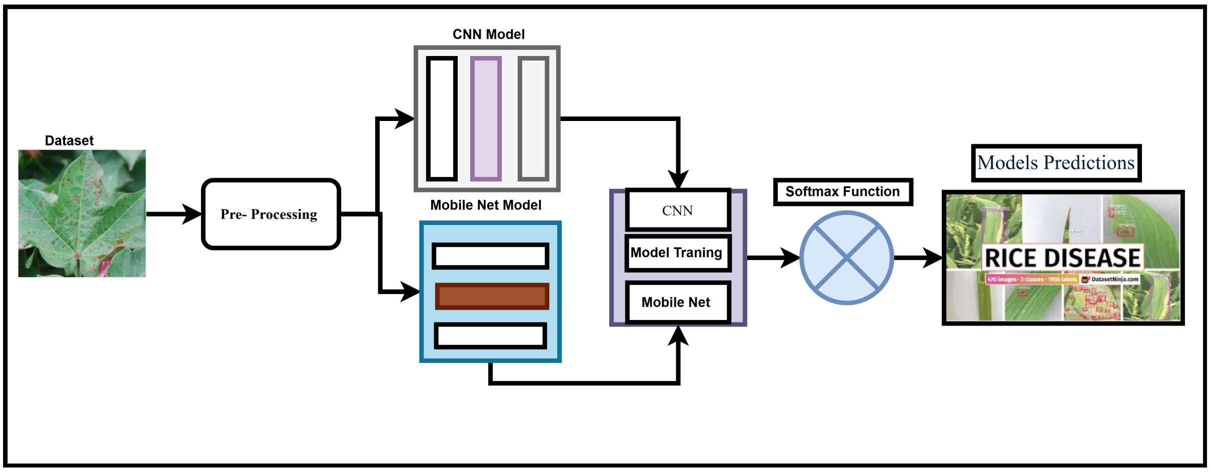

Fig. 3.1. Project Design

3.1. Choice of Method

In this study, we selected image classification methods that were efficient and effective. For disease detection, Convolutional Neural Networks (CNNs) were chosen because of their ability to automatically learn and extract patterns from images. Its lightweight architecture makes it ideal for resource-constrained environments while delivering strong image classification performance.

3.1.1. Justification for the Choice

The methods selected for this study, CNNs and MobileNet, are justified by their effectiveness and efficiency in image classification. CNNs are ideal for learning and extracting meaningful patterns from images, crucial for accurately detecting rice leaf diseases. MobileNet, with its lightweight architecture, offers efficient performance in resource-constrained environments without compromising classification accuracy. Together, these methods ensure high accuracy and optimal resource utilization for disease detection in large datasets.

3.2. Test Strategy

The models were trained on training data and tested on unseen samples using performance metrics such as accuracy, precision, recall, and F1-score.

3.3. Testing and Results

The testing phase evaluated the trained models on unseen data to measure their performance objectively. To compare the effectiveness of CNN and MobileNet models, a variety of metrics were used. Using the evaluation, we gained valuable insight into the strengths and limitations of each method for detecting rice leaf diseases.

3.4. Validation

During training, a validation dataset was used to ensure robustness and reliability of the models, allowing us to fine-tune hyperparameters, prevent overfitting, and improve generalization.

3.5. Ethical, Legal, Social, and Professional Issues

This study adheres to ethical guidelines by utilizing publicly available datasets, ensuring no breach of privacy or data ownership. Legal compliance was maintained by respecting data usage terms. Social considerations were addressed by focusing on applications that enhance food security through disease detection. Professional standards, including transparency and reproducibility, were upheld throughout the research process.

3.6. Practicality

The proposed methodology demonstrates practicality by leveraging lightweight models like MobileNet, which are computationally efficient and suitable for deployment on resource-constrained devices. This enables real-time rice leaf disease detection in agricultural fields, providing a cost-effective solution for farmers to improve crop management and reduce losses.

3.7. Rice Leaf Disease Dataset

This dataset contains 120 high-resolution JPG images of rice leaves precious by three different types of diseasesYusuf et al. 2024. Respectively, the class contains 40 images, providing a composed dataset for training and testing deep learning models. The three sickness categories are dangerous for identifying and sympathetic common threats to rice crops, which are an energy food source for low- and lower-middle-income countries, chiefly in Asia. Detection these diseases early can play a important role in refining crop supervision and food safety.

Dataset Link: https://archive.ics.uci.edu/ml/datasets/Rice+Leaf+Diseases

3.8. Convolutional Neural Networks (CNNs)

A family of deep learning techniques called convolutional neural networks (CNNs) is frequently used to analyse visual input, including pictures or videos. They are very useful for jobs like recognising objects, picture categorisation, and even analysis of video because of automatically learning spatial hierarchies of features Aslam et al. 2019.

3.8.1. Convolutional Layers

CNNs are built on different layers, as shown in Fig. 3.2. The input image (or the previous layer's output) is passed through a series of filters (also called kernels). As the filters slide over the input data, they perform convolution operations, which involve multiplying the filter values by the corresponding input values and summing them. A feature map is created as a result of this operation, which highlights important features in the image, such as edges, textures, and specific shapesO'Shea 2015.

3.8.2. Softmax (for Classification)

In classification problems, a softmax activation is typically used in the final layer. The output is converted into a probability distribution across classes.

3.9. MobileNet Model

The MobileNet deep learning model family is a collection of lightweight deep learning models (Fig. 3.3) In order to solve a wide range of different problems on mobile and edge devices with limited computing resources (like memory and processing power), it is designed for use on mobile devices and edge devices. Despite the fact that the MobileNet is extremely efficient, it performs well on image classification tasks at the same time. Convolutional Neural Networks (CNNs) are used in the MobileNet models to reduce computational costs, making them suitable for resource-constrained environments Y. Chen et al. 2020.

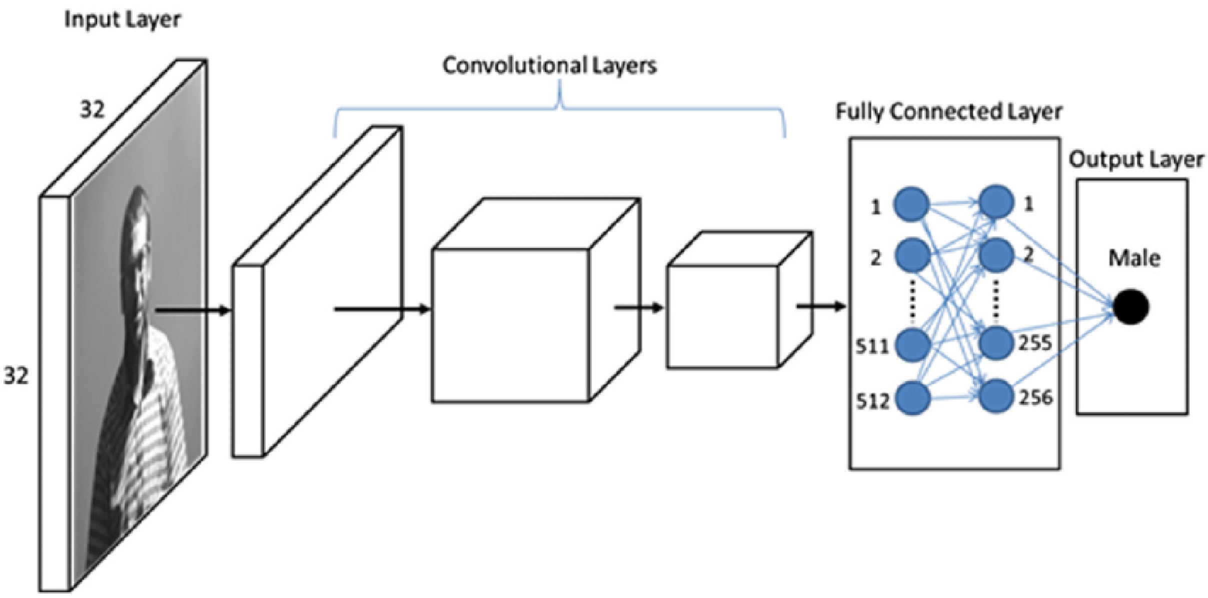

Fig. 3.2. Block Diagram CNN (Aslam et al. 2019)

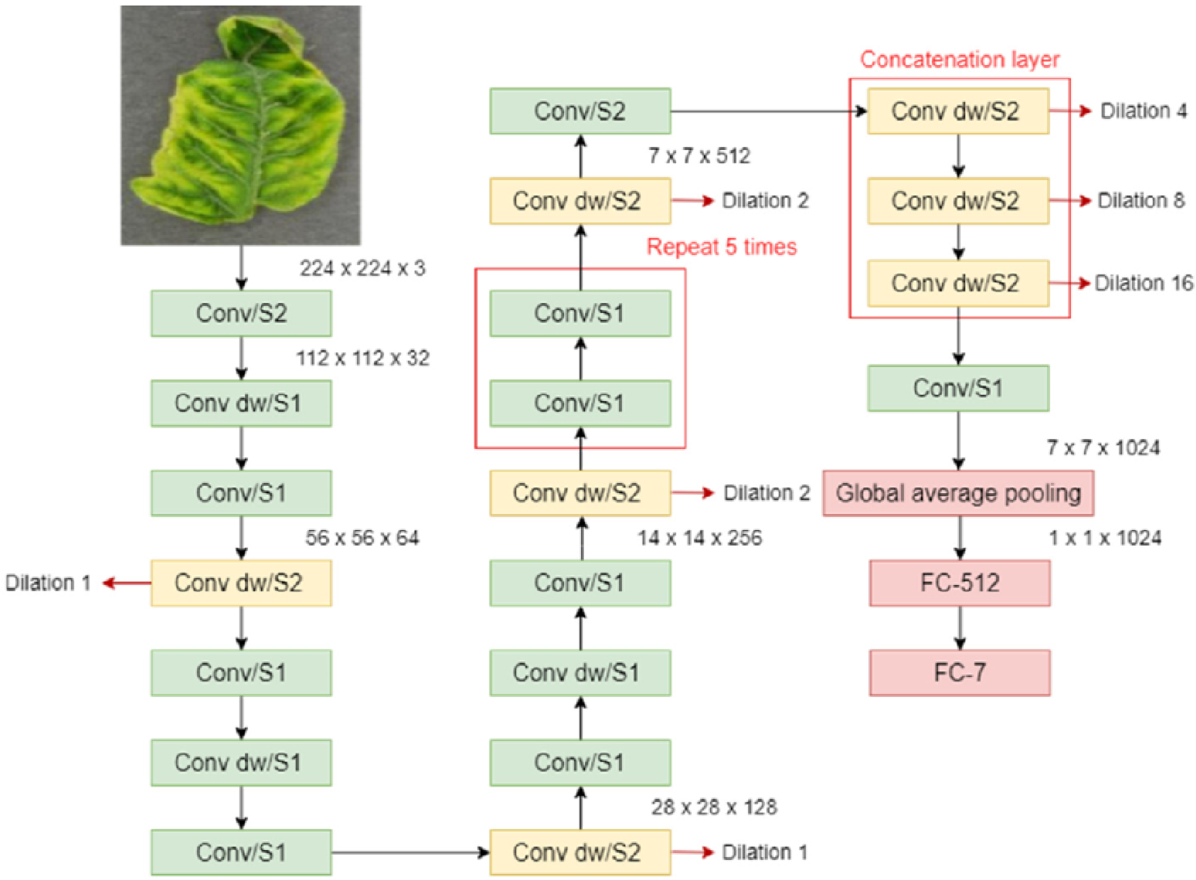

Fig. 3.3. Block Diagram MobileNet-Architecture (Nguyen et al. 2023)

3.10. Dropout Layer

In Dropout layers, neurons are disabled randomly as shown in Fig. 3.4 during training steps in order to prevent overfitting. As a result, the model learns multiple redundant representations of the data instead of relying on just one neuron. As Dropout improves generalization ability, the model is less likely to memorize the training data. Randomly dropping units discourages the network from co-adapting too much, preventing individual neurons from becoming too specialized featuresPark and Kwak 2017.

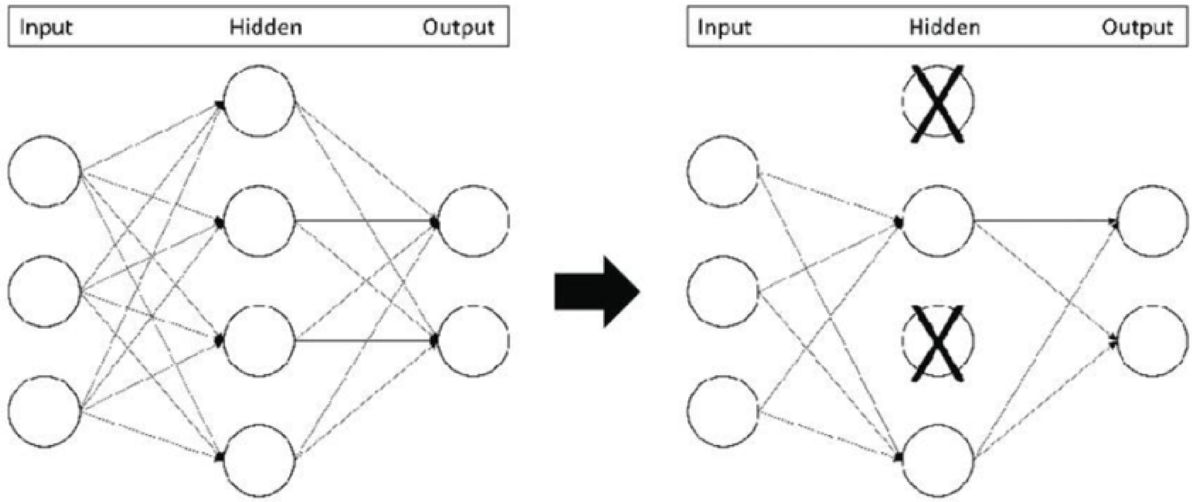

Fig. 3.4. Block Diagram dropout-layer (Al-Galal et al. 2022)

3.11. Performance Evaluation Metrics

Rice Leaf Disease detection models were evaluated using several key metrics: **Accuracy**, **Precision**, **Recall**, and **F1-Score**. Based on these metrics, we are able to determine how well the models perform when it comes to accurately classifying images.

3.11.1. Accuracy

Accuracy measures how many correct predictions were made (both true positives and true negatives) in comparison to total predictions as in Eq. (3.1). Data classification accuracy is a good indicator of a model's performanceStallings and Gillmore 1971.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{3.1}$$

3.11.2. Precision

The precision of the model measures how many true positive predictions it makes and can be calculated through Eq. (3.2). When false positives are expensive, misclassifying a healthy leaf as diseased may result in unnecessary treatments.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.2}$$

3.11.3. Recall

Recall It measures how accurate a model is in identifying actual positive instances as in Eq. (3.3). The metric is useful when it is crucial to detect all positive cases, even at the cost of some false positives (e.g., detecting all diseased leaves to prevent further spread).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.3}$$

3.11.4. F1-Score

The F1-Score provides a balance between precision and recall as shown in Eq. (3.4). It is especially useful when there are fewer positive samples than negative samples..

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.4}$$

4. CHAPTER IV: QUALITY AND RESULTS

The results of the evaluation of CNN Mobile Net, my deep learning model for classifying rice leaf diseases, are presented in this section. For training purposes, categorical cross entropy was used for the loss function. A variety of metrics were used to assess the performance of each model, including accuracy, precession recall, and F 1. Google Collaboratory projects were equipped with 12GB of RAM, 16GB of NVIDIA Tesla T4 GPU RAM, and 2.20GHz Intel(R) Xeon(R) CPUs.

4.1. CNN Model Training and Loss

The deep learning model, trained on 120 high-resolution rice leaf images, achieved 100% accuracy on the training data with a very low loss of 1.7592e-04. However, its performance on the validation dataset was lower, with a 75% accuracy and a validation loss of 2.0851. This indicates good fitting to the training data but difficulty generalizing to unseen data, as shown in Fig4.1 and Fig 4.2. Early disease detection is essential for effective crop management.

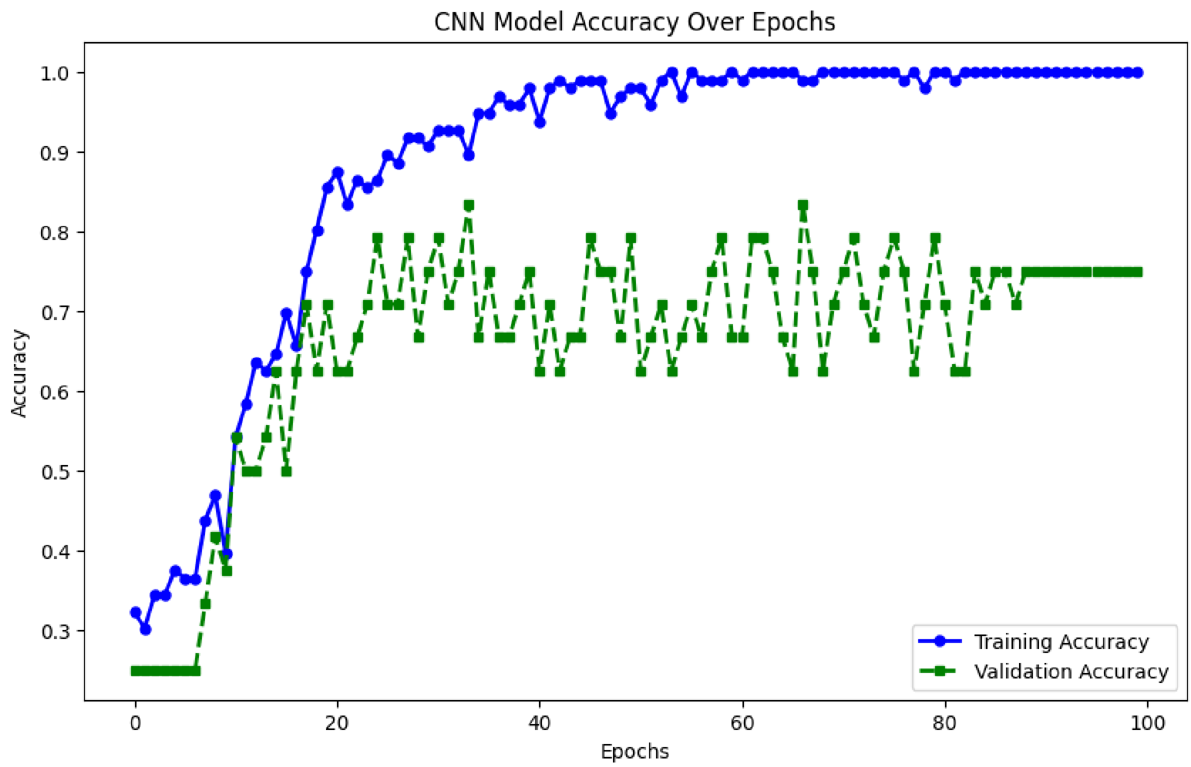

Fig. 4.1. CNN Model Accuracy Over Epochs

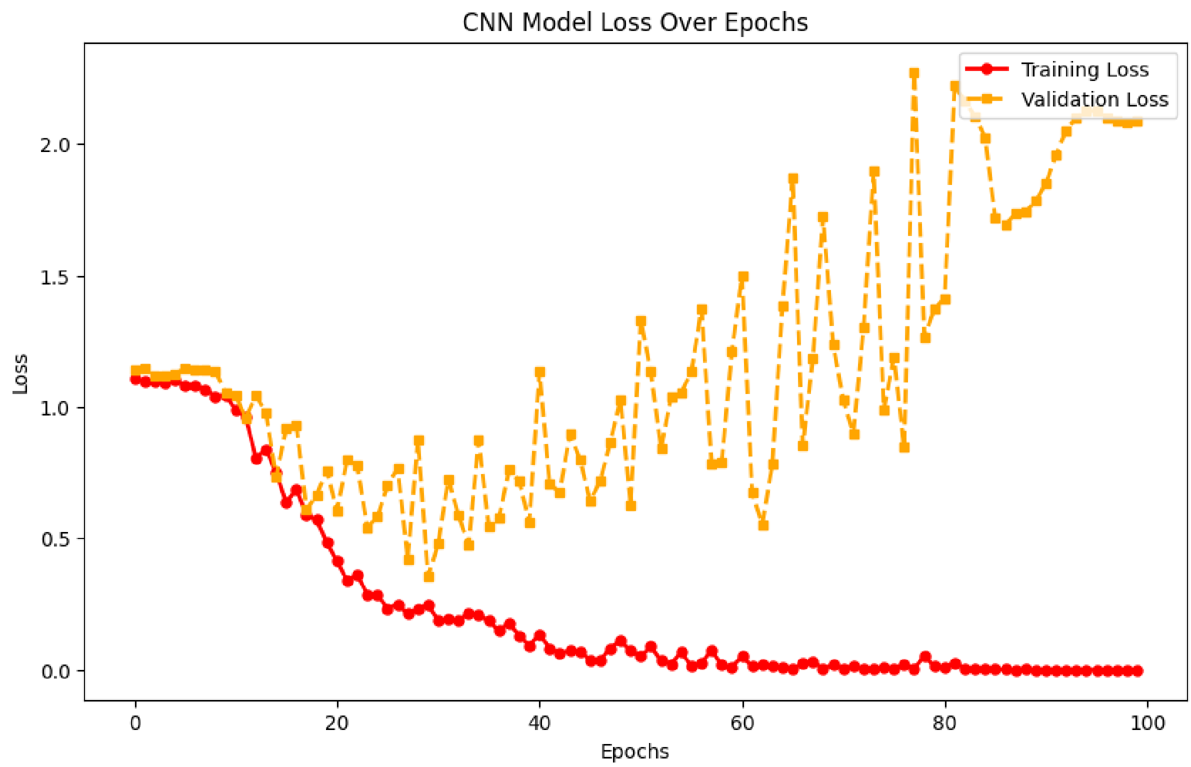

Fig. 4.2. CNN Model Loss Over Epochs

4.2. MobileNet Model Training and Loss

A dataset of 120 high-resolution images of rice leaves was used to train the MobileNet model. Images were classified as Bacterial Leaf Blight, Brown Spot and Leaf Smut. Fig show training and loss 4.3. It is important to emphasize that the model achieved 100% accuracy during training with a loss of only 0.0022, which demonstrated a very effective learning process. Each class contained 40 images.Fig4.4 loss of the model On the validation dataset, MobileNet maintained a high accuracy of 91.67% and a validation loss of 0.5831, showcasing its strong generalization capabilities.

4.2.1. MobileNet model Results

MobileNet was applied to 120 high-resolution images of rice leaf diseases, Bacterial Leaf Blight, Brown Spot, and Leaf Smut 100% training accuracy with a loss of 0.0022. On the validation set, it maintained a strong accuracy of 91.67% with a validation loss of 0.5831. The classification report indicates impressive results across all disease categories, with perfect detection of Bacterial Leaf Blight (precision, recall, and F1-score of 1.00), balanced performance for Brown Spot (precision, recall, and F1-score of 0.91), and strong results for Leaf Smut (precision and recall of 0.86, F1-score of 0.86). With an overall test accuracy of 92%, macro average F1-score of 0.92, and weighted average F1-score of 0.92 as shown in Table 4.1 and Fig. 4.5. The model demonstrates strong generalization and reliability, making MobileNet a

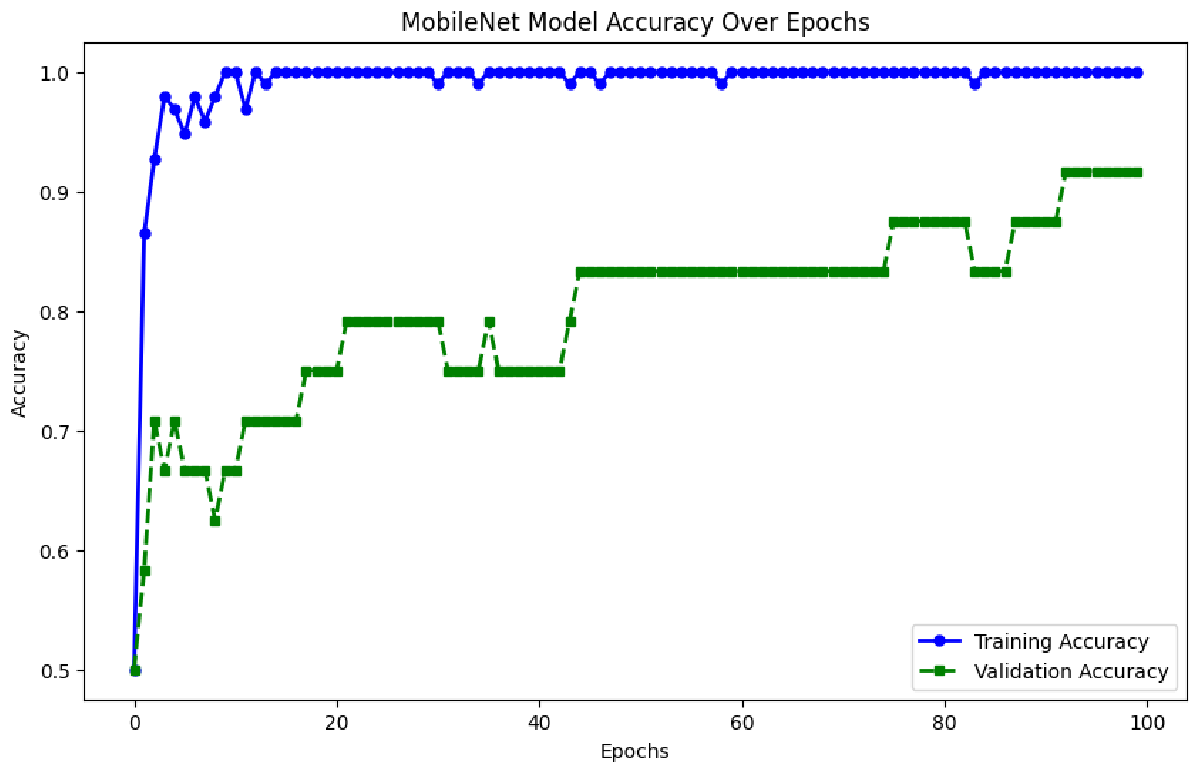

Fig. 4.3. MobileNet Model Accuracy Epochs

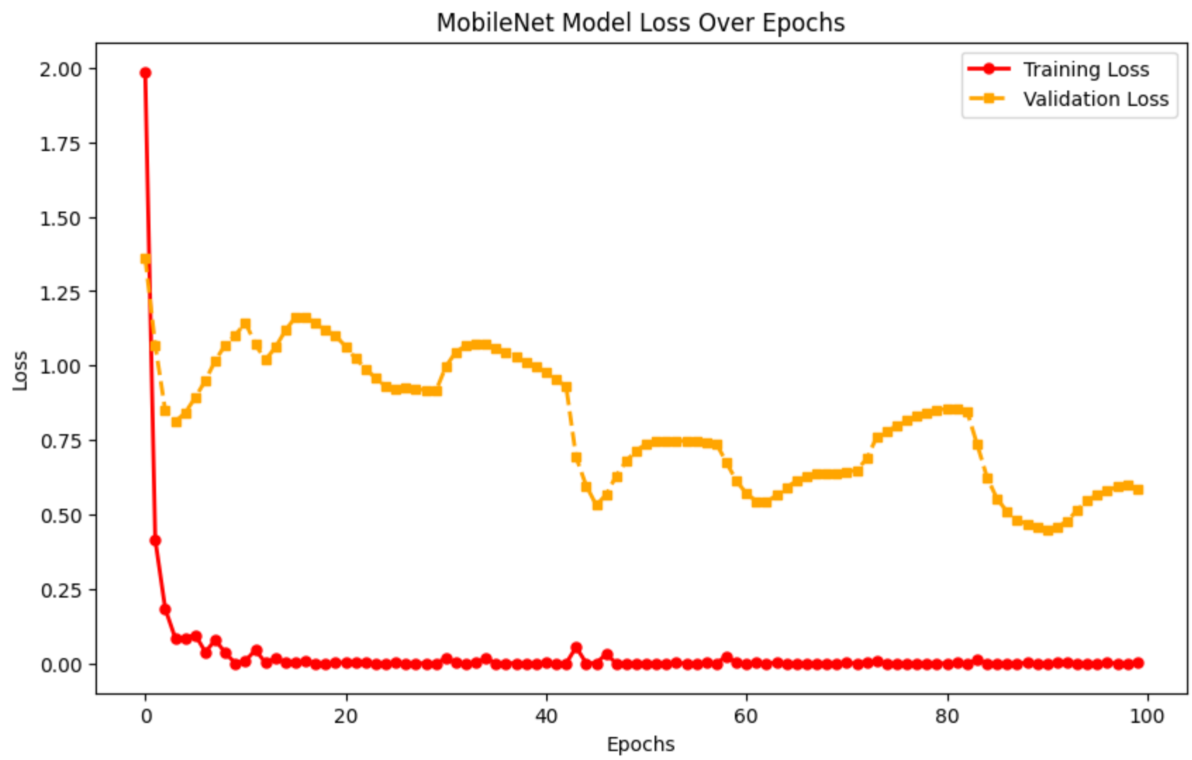

Fig. 4.4. MobileNet Model loss Epochs

promising candidate for real-world crop disease detection applications.

Table 4.1. MobileNet Classification Report

Class	Precision	Recall	F1-Score	Support
Bacterial leaf blight	1.00	1.00	1.00	6
Brown spot	0.91	0.91	0.91	11
Leaf smut	0.86	0.86	0.86	7
Accuracy			0.92	24
Macro avg	0.92	0.92	0.92	24
Weighted avg	0.92	0.92	0.92	24

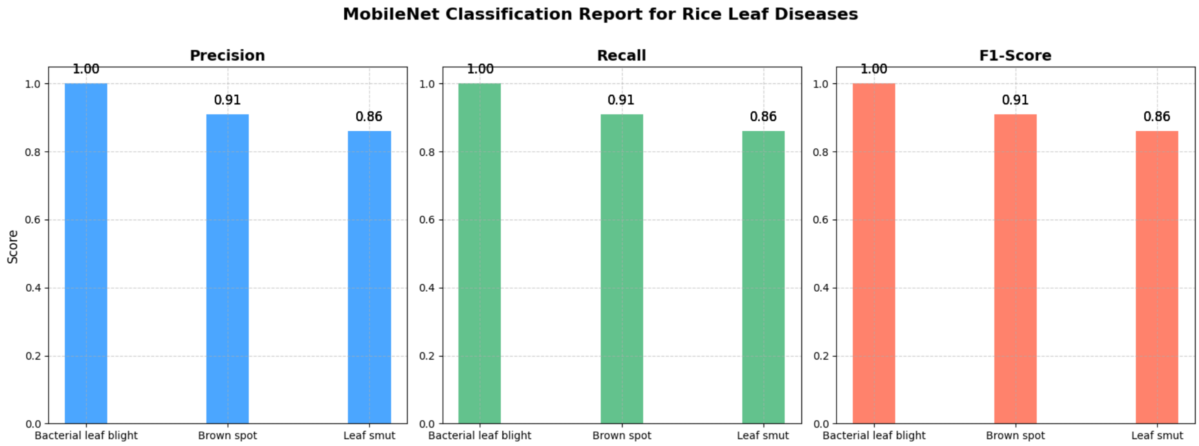

Fig. 4.5. MobileNet Model Results Graph

4.2.2. Confusion Matrix of CNN model

Three rice leaf diseases are predicted by CNN: Bacterial Leaf Blight, Brown Spot, and Leaf Smut. The model correctly classified all six instances of Bacterial Leaf Blight. For Brown Spot, there is some confusion, with one sample misclassified as Bacterial Leaf Blight and four as Leaf Smut, resulting in reduced recall (0.55) as shown in Fig. 4.6. Leaf Smut exhibits better recognition, with only one sample misclassified as Brown Spot, achieving a recall of 0.86.

4.2.3. Confusion matrix of MobileNet

This confusion matrix compares MobileNet with three rice leaf diseases: Bacterial Leaf Blight, Brown Spot, and Leaf Smut. Bacterial Leaf Blight was correctly identified by the model in all six instances as shown in Fig. 4.7. Brown Spot also showed strong performance, with 10 out of 11 samples correctly classified, and only one instance misclassified as Leaf Smut. For Leaf Smut, six samples were correctly classified, but one was misclassified as Brown Spot.

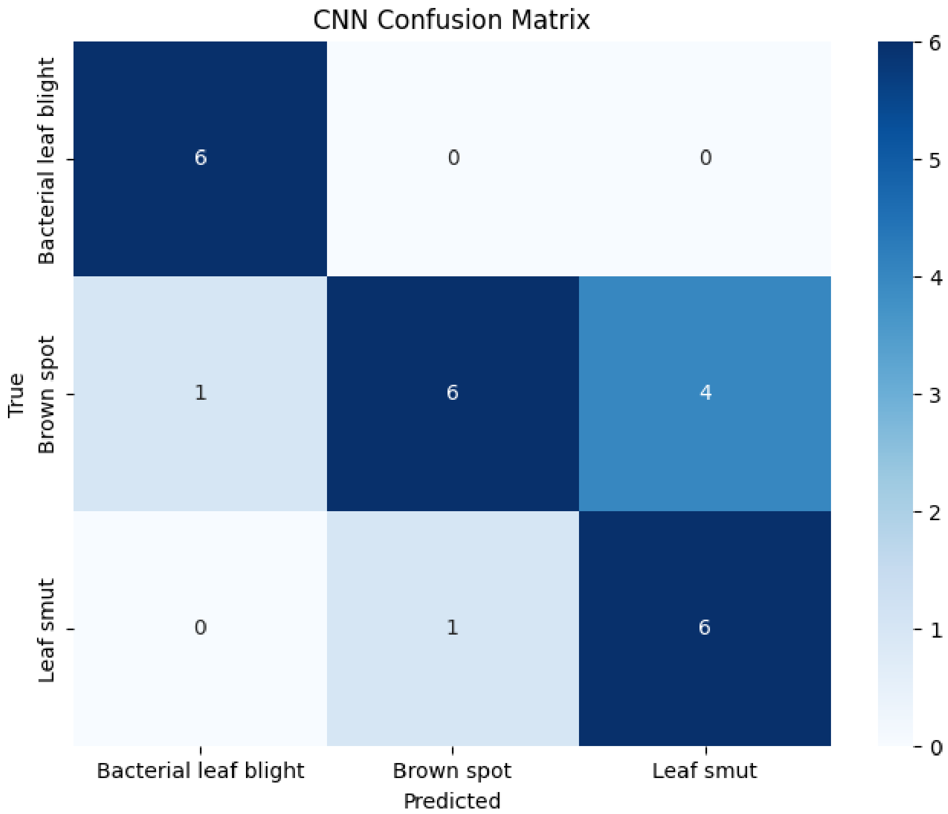

Fig. 4.6. Confusion Matrix of CNN model

Training Performance Both models achieved excellent training accuracy, with the CNN model achieving 100% accuracy and the MobileNet model also reaching 100%. This indicates that both models learned the features of the training data very well. As compared to CNN and MobileNet models as shown in Table4.2 and Fig.4.9, However, the loss values differ significantly, with the CNN model achieving an extremely low training loss of 1.7592e-04, suggesting that it had a very tight fit to the training data. In contrast, the MobileNet model achieved a slightly higher training loss of 0.0022, indicating a slightly less overfitted model.

Validation Performance On the validation set, MobileNet showed superior generalization with a validation accuracy of 91.67%, significantly outperforming the CNN model, which had a validation accuracy of 75%. In this case, CNN overfitted to unseen data, even though it fitted well to the training data. Nevertheless, MobileNet's better performance on the validation set proves that it can generalize well to new, unseen data, making it a more reliable model in practice.

4.2.4. Model Accuracy Comparison for Rice Leaf Disease Detection

The results of the model accuracy comparison for rice leaf disease detection are presented in Fig.4.8 and Table 4.9. The analysis highlights the performance differences among traditional machine learning algorithms, modern convolutional neural networks (CNNs), and proposed

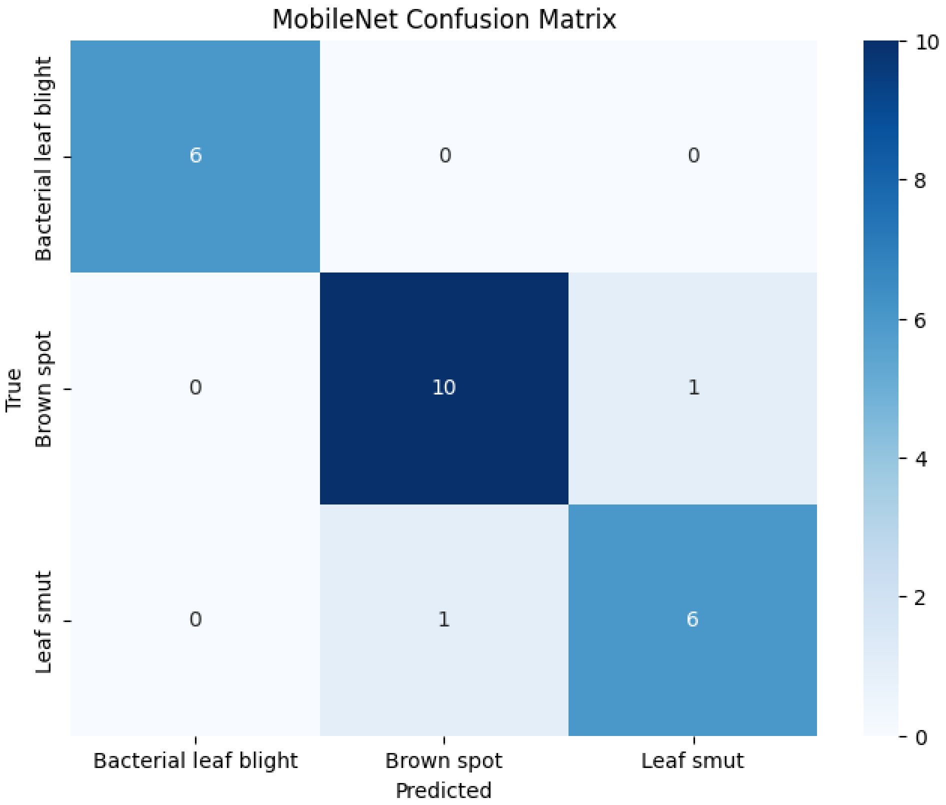

Fig. 4.7. Confusion matrix of MobileNet Model

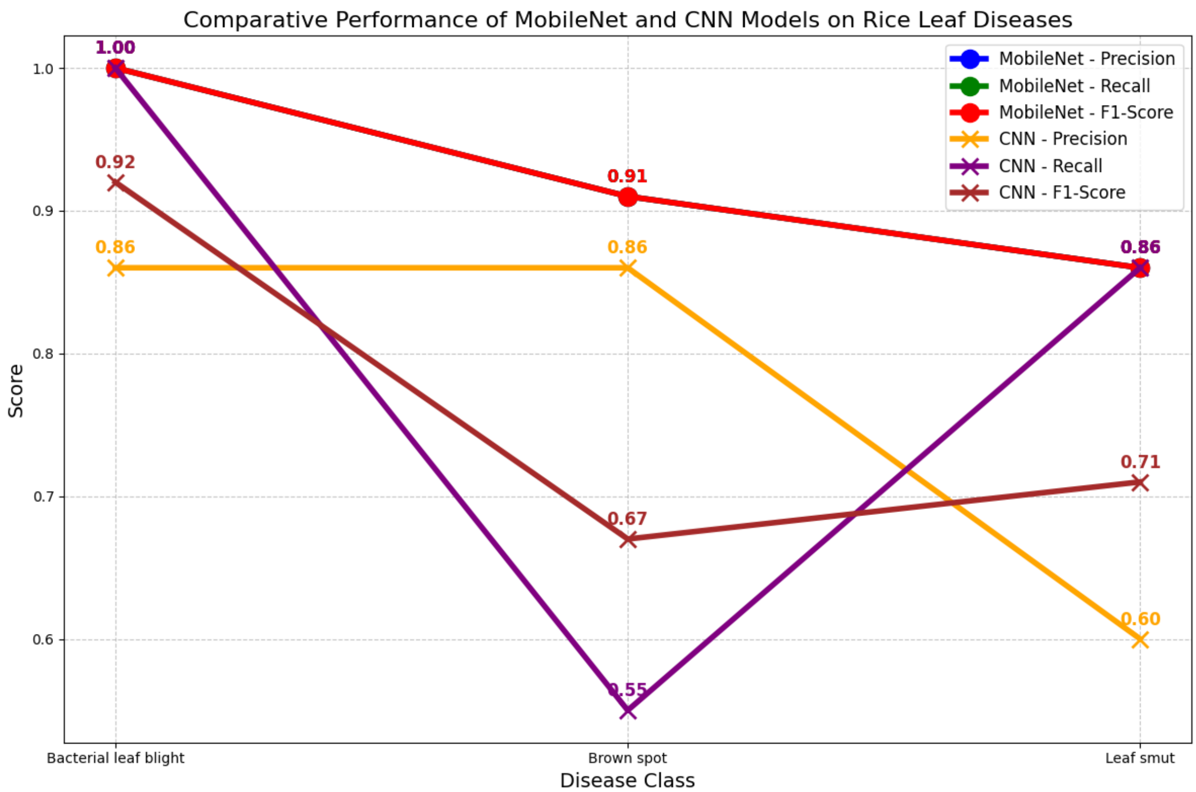

Fig. 4.8. Comparative Classification Results

Table 4.2. Comparative Classification Results for MobileNet and CNN Models on Rice Leaf Diseases

Class	**MobileNet**				**CNN**			
	Precision	**Recall**	**F1-Score**	**Support**	**Precision**	**Recall**	**F1-Score**	**Support**
Bacterial leaf blight	1.00	1.00	1.00	6	0.86	1.00	0.92	6
Brown spot	0.91	0.91	0.91	11	0.86	0.55	0.67	11
Leaf smut	0.86	0.86	0.86	7	0.60	0.86	0.71	7
Accuracy			0.92	24			0.75	24
Macro avg	0.92	0.92	0.92	24	0.77	0.80	0.77	24
Weighted avg	0.92	0.92	0.92	24	0.78	0.75	0.74	24

methodologies Shown in Table4.2.

Table 4.3. Comparison of Models for Rice Leaf Disease Detection

References	**Model Names**	**Accuracy (%)**
Masood et al. 2020	Mask R-CNN	87.6%
Haque et al. 2022	YOLOv5	76.0%
Murugan et al. 2022	Convolutional Neural Network (CNN)	88.28%
Vasantha, Kiranmai, and Krishna 2021	Various Machine Learning Algorithms	90%–91%
Kiratiratanapruk et al. 2022	YOLOv4 with Image Tiling Technique	87.56% to 91.14%
Trinh et al. 2024	Modified YOLOv8	89.9%
Proposed Approach	Mobilenet	92%

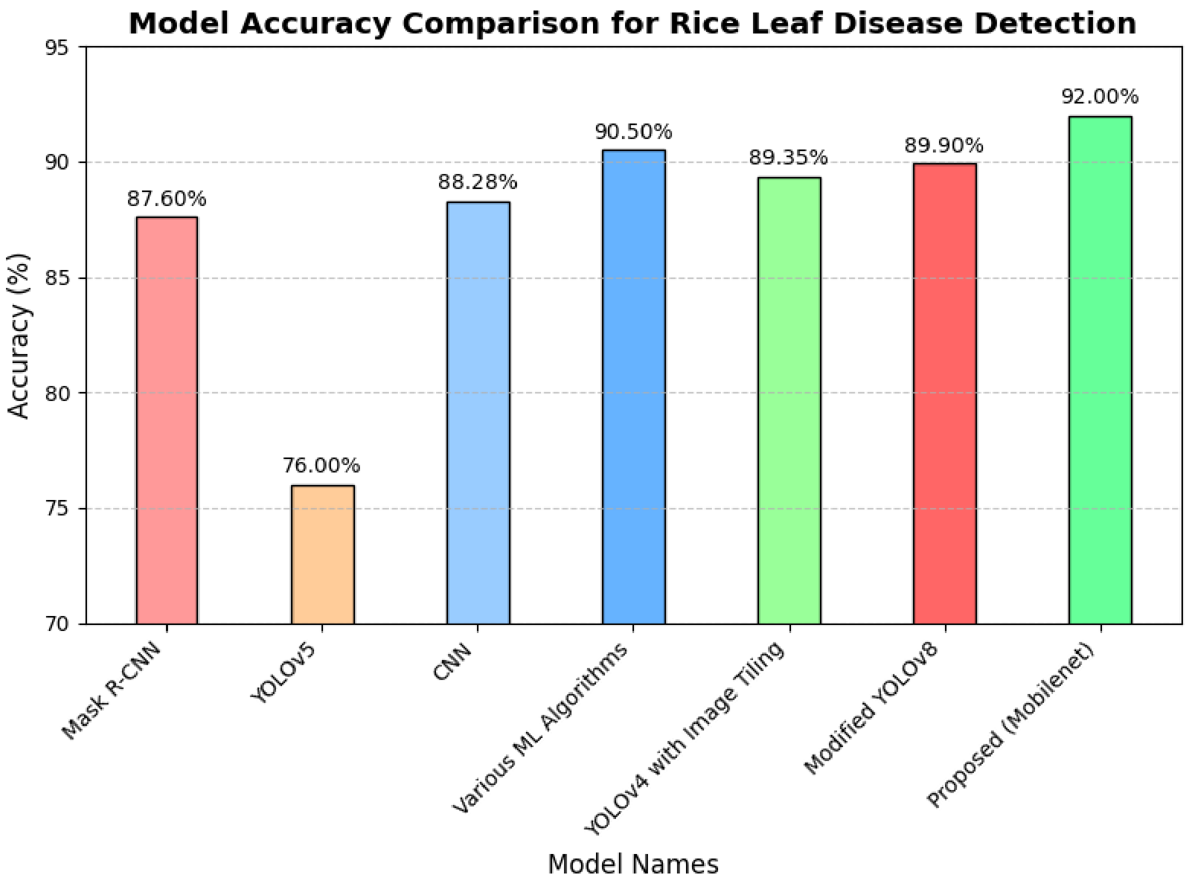

Fig. 4.9. Comparison of Models with Mobil net

5. CHAPTER 5: EVALUATION AND CONCLUSIONS

5.1. Final Evaluation

The experimental results demonstrate that both CNN and MobileNet models performed well in classifying rice leaf diseases, achieving high training accuracy (100%) for both models. MobileNet, MobileNet also outperformed CNN in terms of precision, recall, and F1-score, making it a more reliable model for practical disease detection in rice crops.

5.2. Project Management

The project was executed efficiently using Google Collaboratory with a strong hardware setup (12GB RAM, 16GB GPU RAM, and 2.20GHz Intel CPUs), which enabled the successful training of the CNN and MobileNet models. The task was divided into training, evaluation, and performance analysis phases, ensuring that resources were effectively allocated to enhance model performance. Regular monitoring of model performance and iterative adjustments were made to optimize results.

5.3. Insights Gained

This study highlights the importance of model generalization, as MobileNet's superior performance on the validation set emphasizes its ability to handle unseen data effectively. The challenges of overfitting, particularly in CNN, underscore the need for careful training and validation practices to ensure robust model deployment. Additionally, the importance of dataset diversity and quality is evident in achieving reliable results.

5.4. Comparison to Literature

In comparison to similar studies, MobileNet's strong performance in classification and generalization aligns with findings from other research that emphasizes the efficiency of lightweight models in resource-constrained environments. The CNN model's overfitting issue in this study contrasts with some literature that highlights the potential for CNNs in achieving high accuracy with proper regularization and augmentation techniques.

5.5. Reflection on Challenges

A key challenge in this project was managing overfitting in the CNN model. Despite achieving perfect training accuracy, its lower validation performance highlighted the limitations of its generalization capabilities. Another challenge was ensuring balanced training across all disease categories, as misclassifications in certain classes (such as Brown Spot) impacted overall model performance.

5.6. Conclusion

A comparative analysis of CNN and MobileNet models for rice leaf disease organisation shows their assets and weaknesses. CNN established overfitting contempt high training accuracy, as showed by its meaningfully lower training loss and inferior validation performance. MobileNet, on the other hand, outperformed CNN in terms of validation accuracy and oversimplification, making it more reliable. Also, MobileNet's superior performance across all disease categories, particularly Brown Spot and Leaf Smut, underscores its potential as an effective rice disease detection tool. MobileNet's aptitude to simplify to new data makes it a more healthy solution for automated rice disease diagnosis. With deep learning in agricultural automation, disease detection accuracy can be improved and more efficient and climbable crop management systems can be industrialised.

5.7. Future Work

MobileNet and CNN models could be further assessed by counting more diverse rice diseases and ecological conditions. Incorporating data increase techniques could help alleviate overfitting in CNN models, possibly improving their performance on unseen data. Moreover, exploring hybrid models that combine the strengths of both CNN and MobileNet could lead to even more precise and generalized classification systems. Investigating the integration of these models into real-time agricultural systems, such as mobile apps or drones, could help ease the extensive adoption of automatic disease detection in rice farming. Additional research into transfer learning and fine-tuning of pre-trained models on large-scale agricultural datasets could also enhance the models' efficiency in diverse physical regions. Lastly, the addition of visible and imperceptible symptoms of disease through multi-modal data dispensation (e.g., thermal, multispectral) could recover detection accuracy and enlarge the range of disease diagnoses in precision agriculture.

BIBLIOGRAPHY

Ahmad, Ali et al. (2024). "AI can empower agriculture for global food security: challenges and prospects in developing nations". In: *Frontiers in Artificial Intelligence* 7, p. 1328530.

Amulothu, Durga Venkata Ravi Teja, Rahul R Rodge, Wajid Hasan, and Sheetanshu Gupta (n.d.). "Machine Learning for Pest and Disease Detection in Crops". In: *Agriculture 4.0*. CRC Press, pp. 111–132.

Aslam, Aasma et al. (2019). "Wavelet-based convolutional neural networks for gender classification". In: *Journal of Electronic Imaging* 28.1, pp. 013012–013012.

Awotide, Bola Amoke, Aziz A Karimov, and Aliou Diagne (2016). "Agricultural technology adoption, commercialization and smallholder rice farmers' welfare in rural Nigeria". In: *Agricultural and Food Economics* 4, pp. 1–24.

Balasubramanian, V, Mousa Sie, RJ Hijmans, and Keijiro Otsuka (2007). "Increasing rice production in sub-Saharan Africa: challenges and opportunities". In: *Advances in agronomy* 94, pp. 55–133.

Bandumula, Nirmala (2018). "Rice production in Asia: Key to global food security". In: *Proceedings of the National Academy of Sciences, India Section B: Biological Sciences* 88, pp. 1323–1328.

Bao, Yuequan, Zhiyi Tang, Hui Li, and Yufeng Zhang (2019). "Computer vision and deep learning–based data anomaly detection method for structural health monitoring". In: *Structural Health Monitoring* 18.2, pp. 401–421.

Bari, Bifta Sama et al. (2021). "A real-time approach of diagnosing rice leaf disease using deep learning-based faster R-CNN framework". In: *PeerJ Computer Science* 7, e432.

Buja, Ilaria et al. (2021). "Advances in plant disease detection and monitoring: From traditional assays to in-field diagnostics". In: *Sensors* 21.6, p. 2129.

Chen, Yanjiao, Baolin Zheng, Zihan Zhang, Qian Wang, Chao Shen, and Qian Zhang (2020). "Deep learning on mobile and embedded devices: State-of-the-art, challenges, and future directions". In: *ACM Computing Surveys (CSUR)* 53.4, pp. 1–37.

Deng, Ruoling et al. (2021). "Automatic diagnosis of rice diseases using deep learning". In: *Frontiers in plant science* 12, p. 701038.

Dhaka, Vijaypal Singh, Sangeeta Vaibhav Meena, Geeta Rani, Deepak Sinwar, Muhammad Fazal Ijaz, and Marcin Woźniak (2021). "A survey of deep convolutional neural networks applied for prediction of plant leaf diseases". In: *Sensors* 21.14, p. 4749.

Dhal, Sambandh Bhusan and Debashish Kar (2024). "Transforming Agricultural Productivity with AI-Driven Forecasting: Innovations in Food Security and Supply Chain Optimization". In: *MDPI Forecasting* 6.INL/JOU-24-81560-Rev000.

Fahad, Shah et al. (2019). "Major constraints for global rice production". In: *Advances in rice research for abiotic stress tolerance*. Elsevier, pp. 1–22.

Al-Galal, SABAA AHMED YAHYA, Imad Fakhri Taha Alshaikhli, MM Abdulrazzaq, RAINI Hassan, MM Abdulrazzaq, and HED Moustafa (2022). "Brain Tumor MRI Medical Images

Classification Model Based on CNN (BTMIC-CNN)”. In: *J. Eng. Sci. Technol* 17, pp. 4410–4432.

Haque, Md Ershadul, Ashikur Rahman, Iftekhar Junaeid, Samiul Ul Hoque, and Manoranjan Paul (2022). “Rice leaf disease classification and detection using yolov5”. In: *arXiv preprint arXiv:2209.01579*.

Haridasan, Amritha, Jeena Thomas, and Ebin Deni Raj (2023). “Deep learning system for paddy plant disease detection and classification”. In: *Environmental monitoring and assessment* 195.1, p. 120.

Hasan, Md Mehedi et al. (2023). “Enhancing rice crop management: Disease classification using convolutional neural networks and mobile application integration”. In: *Agriculture* 13.8, p. 1549.

Jafar, Abbas, Nabila Bibi, Rizwan Ali Naqvi, Abolghasem Sadeghi-Niaraki, and Daesik Jeong (2024). “Revolutionizing agriculture with artificial intelligence: plant disease detection methods, applications, and their limitations”. In: *Frontiers in Plant Science* 15, p. 1356260.

John, Matthew Abu, Ibukunoluwa Bankole, Oluwatayo Ajayi-Moses, Tofunmi Ijila, Oluwatimilehin Jeje, and Patil Lalit (2023). “Relevance of advanced plant disease detection techniques in disease and Pest Management for Ensuring Food Security and Their Implication: A review”. In: *American Journal of Plant Sciences* 14.11, pp. 1260–1295.

Kiratiratanapruk, Kantip, Pitchayagan Temniranrat, Wasin Sinthupinyo, Sanparith Marukatat, and Sujin Patarapuwadol (2022). “Automatic detection of rice disease in images of various leaf sizes”. In: *IET Image Processing*.

Kumar, Neeraj et al. (2021). “Challenges and opportunities in productivity and sustainability of rice cultivation system: a critical review in Indian perspective”. In: *Cereal research communications*, pp. 1–29.

Masood, M Hammad, Habiba Saim, Murtaza Taj, and Mian M Awais (2020). “Early disease diagnosis for rice crop”. In: *arXiv preprint arXiv:2004.04775*.

Mohajan, Haradhan (2013). “Food, agriculture and economic situation of Bangladesh”. In.

Mukherjee, Rashmi, Anushri Ghosh, Chandan Chakraborty, Jayanta Narayan De, and Debi Prasad Mishra (2025). “Rice leaf disease identification and classification using machine learning techniques: A comprehensive review”. In: *Engineering Applications of Artificial Intelligence* 139, p. 109639.

Murugan, D et al. (2022). “Paddy Leaf diseases identification on Infrared Images based on Convolutional Neural Networks”. In: *arXiv preprint arXiv:2208.00031*.

Nguyen, Hai Thanh, Huong Hoang Luong, Long Bao Huynh, Bao Quoc Hoang Le, Nhan Hieu Doan, and Duc Thien Dao Le (2023). “An Improved MobileNet for disease detection on tomato leaves”. In: *Advances in Technology Innovation* 8.3, pp. 192–209.

O’Shea, K (2015). “An introduction to convolutional neural networks”. In: *arXiv preprint arXiv:1511.08458*.

Organization, World Health et al. (2013). *Protecting health from climate change: vulnerability and adaptation assessment*. World Health Organization.

Pandian J, Arun, Kanchanadevi K, NR Rajalakshmi, and G Arulkumaran (2022). "An improved deep residual convolutional neural network for plant leaf disease detection". In: *Computational Intelligence and Neuroscience* 2022.1, p. 5102290.

Park, Sungheon and Nojun Kwak (2017). "Analysis on the dropout effect in convolutional neural networks". In: *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part II 13*. Springer, pp. 189–204.

Prajapati, Harshadkumar B, Jitesh P Shah, and Vipul K Dabhi (2017). "Detection and classification of rice plant diseases". In: *Intelligent Decision Technologies* 11.3, pp. 357–373.

Ramesh, S and D Vydeki (2020). "Recognition and classification of paddy leaf diseases using Optimized Deep Neural network with Jaya algorithm". In: *Information processing in agriculture* 7.2, pp. 249–260.

Razzak, Muhammad Imran, Saeeda Naz, and Ahmad Zaib (2018). "Deep learning for medical image processing: Overview, challenges and the future". In: *Classification in BioApps: Automation of decision making*, pp. 323–350.

Roy, Arunabha M and Jayabrata Bhaduri (2021). "A deep learning enabled multi-class plant disease detection model based on computer vision". In: *Ai* 2.3, pp. 413–428.

Saberi Anari, Maryam (2022). "A Hybrid Model for Leaf Diseases Classification Based on the Modified Deep Transfer Learning and Ensemble Approach for Agricultural AIoT-Based Monitoring". In: *Computational Intelligence and Neuroscience* 2022.1, p. 6504616.

Sethy, Prabira Kumar, Nalini Kanta Barpanda, Amiya Kumar Rath, and Santi Kumari Behera (2020a). "Deep feature based rice leaf disease identification using support vector machine". In: *Computers and Electronics in Agriculture* 175, p. 105527.

— (2020b). "Image processing techniques for diagnosing rice plant disease: a survey". In: *Procedia Computer Science* 167, pp. 516–530.

Singh, Kaushik Kunal (2018). "An artificial intelligence and cloud based collaborative platform for plant disease identification, tracking and forecasting for farmers". In: *2018 IEEE international conference on cloud computing in emerging markets (CCEM)*. IEEE, pp. 49–56.

Stallings, William M and Gerald M Gillmore (1971). "A note on "accuracy" and "precision"". In: *Journal of Educational Measurement* 8.2, pp. 127–129.

Strange, Richard N and Peter R Scott (2005). "Plant disease: a threat to global food security". In: *Annu. Rev. Phytopathol.* 43.1, pp. 83–116.

Trinh, Dong Cong, Anh Tuan Mac, Khanh Giap Dang, Huong Thanh Nguyen, Hoc Thai Nguyen, and Thanh Dang Bui (2024). "Alpha-EIOU-YOLOv8: an improved algorithm for rice leaf disease detection". In: *AgriEngineering* 6.1, pp. 302–317.

Upadhyay, Santosh Kumar and Avadhesh Kumar (2022). "A novel approach for rice plant diseases classification with deep convolutional neural network". In: *International Journal of Information Technology* 14.1, pp. 185–199.

Vasantha, Sandhya Venu, Bejjam Kiranmai, and S Rama Krishna (2021). "Techniques for rice leaf disease detection using machine learning algorithms". In: *Int. J. Eng. Res. Technol* 9.8, pp. 162–166.

Velusamy, Durgadevi, A Mounika, D Ragulraj, and K Vigneshwaran (2023). "Automatic Disease Classification of Paddy Leaf Images Using Deep Learning Algorithms". In: *2023 4th International Conference on Signal Processing and Communication (ICSPC)*. IEEE, pp. 325–329.

Wani, Javaid Ahmad, Sparsh Sharma, Malik Muzamil, Suhaib Ahmed, Surbhi Sharma, and Saurabh Singh (2022). "Machine learning and deep learning based computational techniques in automatic agricultural diseases detection: Methodologies, applications, and challenges". In: *Archives of Computational methods in Engineering* 29.1, pp. 641–677.

Yusuf, Hassan Muhammad, Yusuf Sahabi Ali, Amina Hassan Abubakar, Mohammed Abdullahi, and Ibrahim Hayatu Hassan (2024). "A systematic review of deep learning techniques for rice disease recognition: Current trends and future directions". In: *Franklin Open*, p. 100154.

Zhao, Mingchao, Yongjun Lin, and Hao Chen (2020). "Improving nutritional quality of rice for human health". In: *Theoretical and Applied Genetics* 133, pp. 1397–1413.

Zhou, Changjian, Yujie Zhong, Sihan Zhou, Jia Song, and Wensheng Xiang (2023). "Rice leaf disease identification by residual-distilled transformer". In: *Engineering Applications of Artificial Intelligence* 121, p. 106020.

APPENDIX

```
import numpy as np
import mataplotlib.pyplot as plt
import seaborn as sns
import tensorflow as tf
from sklearn.metrcis import
classification_report, confusion_matrix, accuracy_score, precision_score
    , recall_score, f1_score
from tensorflow.keras.utils import to_cateorical
from tensorflow.keras.callbacks import ModelCheckpoint
import os
import cv2
import albumentations as albu
from albumentations import Compose, ShiftScaleRotate, Resize
from sklearn.utils import shuffle
from keras.models import Sequential
from sklearn.model_selection import train_test_split
from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
from tensorflow.keras.applications import MobileNet
from tesndorflow.keras.layer import
GlobalAveragePooling2D, BatchNormalization
from tensorflow.keras.optimizers import adam


# Setup Environment
input_shape_2D = (224, 224) # For 2D images
input_shape_3D = (224, 224, 3) # For RGB images
seed = 1
batch_size = 32
epochs = 100

from google.colab import drive
drive.mount('/content/drive')

# Load Image data
data = tf.keras.utils.image_dataset_from_directory(
    directory="/content/drive/MyDrive/rice_leaf_diseases", # Adjust path
        as necessary
    labels='inferred', label_mode='int', class_names=None, color_mode='
       rgb',
    image_size=(224, 224), seed=1
)

```

```
# Print class names
class_names = data.class_names
class_names

# Visualize sample images
plt.figure(figsize=(10, 10))
for images, labels in data.take(1):
    for i in range(25):
        plt.subplot(5, 5, i + 1)
        plt.imshow(images[i].numpy().astype('uint8'))
        plt.title(class_names[labels[i]])
        plt.axis('off')
plt.tight_layout()
plt.show()

# Data Preprocessing
# Prepare data for training and testing
X = []
y = []
for images, labels in data:
    X.append(images.numpy())
    y.append(labels.numpy())

X = np.concatenate(X, axis=0)
y = np.concatenate(y, axis=0)

# Normalize images
X = X.astype('float32') / 255

# Use train_test_split to randomly split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
    random_state=42)

# Convert labels to one-hot encoding
y_train = to_categorical(y_train, len(class_names))
y_test = to_categorical(y_test, len(class_names))

# Build CNN Model
model_cnn = Sequential()
model_cnn.add(Conv2D(filters=16, kernel_size=3, padding='same', strides
    =1, activation='relu', input_shape=(224, 224, 3)))
model_cnn.add(MaxPooling2D(pool_size=(3, 3)))

model_cnn.add(Conv2D(filters=32, kernel_size=3, padding='same', strides
    =1, activation='relu'))
model_cnn.add(MaxPooling2D(pool_size=(3, 3)))

model_cnn.add(Conv2D(filters=64, kernel_size=3, padding='same', strides
    =1, activation='relu'))
model_cnn.add(MaxPooling2D(pool_size=(3, 3)))
```

```

model_cnn.add(Conv2D(filters=128, kernel_size=3, padding='same', strides=1, activation='relu'))
model_cnn.add(MaxPooling2D(pool_size=(2, 2)))

model_cnn.add(Conv2D(filters=256, kernel_size=3, padding='same', strides=1, activation='relu'))
model_cnn.add(MaxPooling2D(pool_size=(2, 2)))

model_cnn.add(Dropout(0.3))
model_cnn.add(Flatten())
model_cnn.add(Dropout(0.5))
model_cnn.add(Dense(250, activation='relu'))
model_cnn.add(Dense(100, activation='relu'))
model_cnn.add(Dense(len(class_names), activation='softmax'))

model_cnn.summary()

# Compile the CNN model
model_cnn.compile(optimizer='Adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Define ModelCheckpoint callback
check_pointer = ModelCheckpoint(
    filepath='cnn_model.best.weights.h5', monitor='val_accuracy', save_best_only=True,
    mode='auto', save_weights_only=True
)

# Train CNN model
history_cnn = model_cnn.fit(
    X_train, y_train, batch_size=batch_size, epochs=epochs, validation_data=(X_test, y_test),
    callbacks=[check_pointer], verbose=1
)

# Visualize training and validation accuracy
plt.figure(figsize=(10, 6))
plt.plot(history_cnn.history['accuracy'], label='Training Accuracy', color='blue', linewidth=2, marker='o', markersize=5)
plt.plot(history_cnn.history['val_accuracy'], label='Validation Accuracy', color='green', linewidth=2, linestyle='--', marker='s', markersize=5)
plt.title("CNN Model Accuracy Over Epochs")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend(loc="lower right")
plt.show()

# Visualize training and validation loss
```

```
plt.figure(figsize=(10, 6))
plt.plot(history_cnn.history['loss'], label='Training Loss', color='red'
    , linewidth=2, marker='o', markersize=5)
plt.plot(history_cnn.history['val_loss'], label='Validation Loss', color
    ='orange', linewidth=2, linestyle='--', marker='s', markersize=5)
plt.title("CNN Model Loss Over Epochs")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.legend(loc="upper right")
plt.show()
```

```

# Predictions for CNN model
y_pred_cnn = model_cnn.predict(X_test)
predicted_classes_cnn = np.argmax(y_pred_cnn, axis=1)

# Classification Report for CNN
print("CNN Classification Report:\n", classification_report(np.argmax(
    y_test, axis=1), predicted_classes_cnn, target_names=class_names))

# Confusion Matrix for CNN
cm_cnn = confusion_matrix(np.argmax(y_test, axis=1),
    predicted_classes_cnn)
plt.figure(figsize=(8, 6))
sns.heatmap(cm_cnn, annot=True, fmt='d', cmap='Blues', xticklabels=
    class_names, yticklabels=class_names)
plt.title('CNN Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('True')
plt.show()
```

```
plt.show()

# Predictions and Evaluation for MobileNet
y_pred_mobilenet = model_mobilenet.predict(X_test)
predicted_classes_mobilenet = np.argmax(y_pred_mobilenet, axis=1)

# Classification Report for MobileNet
print("MobileNet Classification Report:\n", classification_report(np.
    argmax(y_test, axis=1), predicted_classes_mobilenet, target_names=
    class_names))

# Confusion Matrix for MobileNet
cm_mobilenet = confusion_matrix(np.argmax(y_test, axis=1),
    predicted_classes_mobilenet)
plt.figure(figsize=(8, 6))
sns.heatmap(cm_mobilenet, annot=True, fmt='d', cmap='Blues', xticklabels
    =class_names, yticklabels=class_names)
plt.title('MobileNet Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('True')
plt.show()
```

```python
# Calculate accuracy, precision, recall, F1 score
accuracy_cnn = accuracy_score(np.argmax(y_test, axis=1),
    predicted_classes_cnn)
precision_cnn = precision_score(np.argmax(y_test, axis=1),
    predicted_classes_cnn, average='weighted')
recall_cnn = recall_score(np.argmax(y_test, axis=1),
    predicted_classes_cnn, average='weighted')
f1_cnn = f1_score(np.argmax(y_test, axis=1), predicted_classes_cnn,
    average='weighted')

print(f"CNN Accuracy: {accuracy_cnn}")
print(f"CNN Precision: {precision_cnn}")
print(f"CNN Recall: {recall_cnn}")
print(f"CNN F1 Score: {f1_cnn}")

# Load pre-trained MobileNet model (without top classification layers)
base_model = MobileNet(weights='imagenet', include_top=False,
    input_shape=(224, 224, 3))

# Freeze the layers of the base model
```

```python
base_model.trainable = False

# Add custom layers on top of MobileNet
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(1024, activation='relu')(x)
x = BatchNormalization()(x)
x = Dropout(0.5)(x)
x = Dense(512, activation='relu')(x)
x = Dropout(0.5)(x)
x = Dense(len(class_names), activation='softmax')(x)

# Define the MobileNet model
model_mobilenet = Model(inputs=base_model.input, outputs=x)

# Compile the MobileNet model
model_mobilenet.compile(optimizer=Adam(), loss='categorical_crossentropy
    ', metrics=['accuracy'])

# Train MobileNet model
history_mobilenet = model_mobilenet.fit(
    X_train, y_train, batch_size=32, epochs=100,
    validation_data=(X_test, y_test), callbacks=[check_pointer], verbose
        =1
)
```

```
base_model.trainable = False

# Add custom layers on top of MobileNet
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(1024, activation='relu')(x)
x = BatchNormalization()(x)
x = Dropout(0.5)(x)
x = Dense(512, activation='relu')(x)
x = Dropout(0.5)(x)
x = Dense(len(class_names), activation='softmax')(x)

# Define the MobileNet model
model_mobilenet = Model(inputs=base_model.input, outputs=x)

# Compile the MobileNet model
model_mobilenet.compile(optimizer=Adam(), loss='categorical_crossentropy
    ', metrics=['accuracy'])

# Train MobileNet model
history_mobilenet = model_mobilenet.fit(
    X_train, y_train, batch_size=32, epochs=100,
    validation_data=(X_test, y_test), callbacks=[check_pointer], verbose
        =1
)

```

```
# Visualize training and validation accuracy for MobileNet
plt.figure(figsize=(10, 6))
plt.plot(history_mobilenet.history['accuracy'], label='Training Accuracy
    ', color='blue', linewidth=2, marker='o', markersize=5)
plt.plot(history_mobilenet.history['val_accuracy'], label='Validation
    Accuracy', color='green', linewidth=2, linestyle='--', marker='s',
    markersize=5)
plt.title("MobileNet Model Accuracy Over Epochs")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend(loc="lower right")
plt.show()

# Visualize training and validation loss for MobileNet
plt.figure(figsize=(10, 6))
plt.plot(history_mobilenet.history['loss'], label='Training Loss', color
    ='red', linewidth=2, marker='o', markersize=5)
plt.plot(history_mobilenet.history['val_loss'], label='Validation Loss',
     color='orange', linewidth=2, linestyle='--', marker='s', markersize
    =5)
plt.title("MobileNet Model Loss Over Epochs")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.legend(loc="upper right")
```