



RICE DISEASE IDENTIFICATION THROUGH TRANSFER LEARNING

SUBMITTED BY

Umme Sadia Raisa

Roll No: MUH1911006F

Session: 2018-19

Year 04, Term 02

Department of Information & Communication Engineering

THESIS SUPERVISOR

Prof. Dr. Md. Ashikur Rahman Khan

Professor

Department of Information & Communication Engineering

**DEPARTMENT OF INFORMATION AND COMMUNICATION ENGINEERING
NOAKHALI SCIENCE AND TECHNOLOGY UNIVERSITY**

May 2024

ACCEPTANCE

I hereby declare that I have checked this thesis and, in my opinion, this thesis is adequate in terms of scope and quality for the degree of B.Sc. (Eng.) in Information and Communication Engineering under the course entitled “ICE-4218”.

Prof. Dr. Md. Ashikur Rahman Khan

Professor

Department of Information & Communication Engineering

Noakhali Science and Technology University

Noakhali – 3814

DECLARATION

This thesis report is submitted to the Department of Information and Communication Engineering (ICE), Noakhali Science and Technology University, Noakhali-3814, in partial fulfillment of requirements for the degree of B.Sc. (Eng.) in Information and Communication Engineering under the course entitled “ICE-4218”.

I hereby declare that this thesis proposal has not been submitted elsewhere for the requirement of any degree, diploma, or publication.

Umme Sadia Raisa

Roll No: MUH1911006F

Session: 2018-19

Year: 4, Term: 2

Department of Information and Communication Engineering (ICE)

Noakhali Science and Technology University

Noakhali-3814.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my thesis supervisor, Prof. Dr. Md. Ashikur Rahman Khan, for his invaluable guidance, continuous encouragement, and constant support in making this research possible. I am also thankful to my friends for their constructive feedback and encouragement. I appreciate his constant support from the first day I applied to the graduate program to these concluding moments. I am truly grateful for his progressive vision for my training in science, his tolerance of my naïve mistakes, and his progressive vision for my future career. This thesis would not have been possible without the assistance and cooperation of Noakhali Science and Technology University and the participants who generously contributed their time and insights. I acknowledge my sincere indebtedness and gratitude to my parents for their love dream and sacrifice throughout my life—special thanks to my friends who constantly encouraged and supported me during this challenging time.

ABSTRACT

In the rural landscape of Bangladesh, where rice reigns supreme as the cornerstone of sustenance, disease poses a formidable threat to the nation's agricultural backbone. With a staggering 78% of the country's arable land dedicated to rice cultivation, the specter of crop diseases looms large, exacting a heavy toll on farmers' livelihoods and the nation's food security. Efficient disease detection is paramount to safeguarding rice yields and ensuring food security. Traditionally, this task has been labor-intensive and time-consuming, demanding extensive manual effort. However, the advent of automated systems offers a beacon of hope in this agricultural conundrum. Hence deep learning is getting popular for its transformative technology rapidly eclipsing conventional approaches with its unparalleled performance. The objectives of this research were twofold: to achieve higher recognition accuracy for rice diseases through transfer learning and to compare the performance of the three CNN architectures. The study found that VGGNet16 and InceptionResNetV2 models, when optimized with RMSprop, achieved high accuracy rates of 98.44% and 98.21%, respectively, demonstrating their robustness and reliability for rice disease detection. In contrast, ResNet50 showed lower performance with an accuracy of 50.99%, indicating its limitations in this specific application. This research contributes by demonstrating the effectiveness of transfer learning in improving disease detection accuracy and providing a comparative analysis of CNN models and optimizers. The findings highlight the potential of these models for broader agricultural applications, offering insights into crop health management beyond rice cultivation. The study underscores the importance of selecting appropriate models and optimization strategies to enhance the efficiency and accuracy of automated disease detection systems, thereby supporting agricultural sustainability and food security.

KEYWORDS: Rice; Disease Classification; CNN; Deep Learning; Transfer Learning; VGGNet; ResNet50; InceptionResNetV2; Feature Extraction;

TABLE OF CONTENTS

	Page
Title Page	i
Acceptance	ii
Declaration	iii
Acknowledgment	iv
Abstract	v
Table of Contents	vi
List of Tables	x
List of Figures	xi
Chapter 1 Introduction	1-5
1.1 Introduction	1
1.2 Problem Statement	2
1.3 Motivation for this Study	3
1.4 Research Objectives	4
1.5 Contribution of this Study	4
1.6 Scope	5
1.7 Research Outlines	5
Chapter 2 Literature Review	6-15

2.1	Introduction	6
2.2	Objectives of the Review	7
2.3	Types of Rice Diseases	7
2.4	Distinct Neural Networks for Rice Disease Prediction	9
2.5	Different Convolution Neural Networks and their Scope	13
2.6	Research Gap	14
2.7	Summary	15
Chapter 3	Methodology	16-37
3.1	Introduction	16
3.2	Model Development	16
3.2.1	Neural Network Models	17
a)	Convolution Neural Network	17
b)	VGGNet16	18
c)	ResNet50	19
d)	InceptionResNetV2	20
3.2.2	Model Architecture	20
a)	CNN Architecture	21
b)	VGGNet16 Architecture	22
c)	ResNet50 Architecture	23
d)	InceptionResNetV2 Architecture	23
3.3	Justification for Model Selection	24

3.4	Transfer Learning Framework	26
3.5	Workflow	27
3.6	Dataset Preparation	29
3.6.1	Dataset Acquisition	29
3.6.2	Image Resize & Rescaling	29
3.7	Training and Testing the Models	30
3.7.1	Training and Testing Dataset Split	30
3.7.2	Training Phase	31
3.7.3	Optimization	31
3.7.4	Testing Phase	32
3.8	Measurement Factors	33
i.	Confusion Matrix	33
ii.	Accuracy	34
iii.	Precision	34
iv.	Recall	35
v.	F1 Score	35
vi.	Loss Calculation	35
3.9	Implementation Tools	36
3.10	Summary	37
Chapter 4 Result Analysis & Discussion		38-55
4.1	Introduction	38

4.2	Performance of the Neural Network Models	38
4.2.1	VGGNet16	39
4.2.2	ResNet50	41
4.2.3	InceptionResNetV2	44
4.3	Optimizer	46
4.3.1	Accuracy Calculation	46
4.3.2	Error Analysis	48
4.4	Comparative Analysis	49
4.5	Testing for Different Crops	50
a)	VGGNet16	50
b)	ResNet50	51
c)	InceptionResNetV2	52
4.6	Comparison with Existing Works	54
4.7	Summary	55
Chapter 5	Conclusion	56-57
5.1	Conclusion	56
5.2	Future Scope	56
5.3	Summary	57
References		58-60

LIST OF TABLES

Table No.	Title	Page
2.1	Types of Rice Diseases	7
2.2	Related works and their dissimilarities with this study	11
3.1	Model Architecture of CNN	21
3.2	Model Architecture of VGGNet16	22
3.3	Model Architecture of ResNet50	24
3.4	Model Architecture of InceptionResNetV2	25
3.5	Train Split Ratio	31
4.1	Error Summary	48
4.2	Models Comparison Parameters	49
4.3	Confusion matrix for different crops using vggnet16	50
4.4	Confusion matrix for different crops using resnet50	51
4.5	Confusion matrix for different crops using inceptionresnetv2	53
4.6	Accuracy summary for different crops	53
4.7	Comparison with existing works	54

LIST OF FIGURES

Figure No.	Title	Page
3.1	CNN model architecture	18
3.2	VGGNet 16 model architecture	19
3.3	ResNet50 model architecture	19
3.4	InceptionResNetV2 model architecture	20
3.5	Transform Learning Framework	27
3.6	Overview of the Workflow	28
3.7	Sample images of rice leaf dataset	29
4.1	Confusion matrix for vggnet16	39
4.2	Confusion matrix for resnet50	42
4.3	Confusion matrix for inceptionresnetv2	44
4.4	Performance analysis for different optimizers	47
4.5	Impact of Optimizer Usage on Model Performance	48

CHAPTER 1

INTRODUCTION

1.1 Introduction

The third most prevalent cereal grain, rice (*oryza sativa*) is a staple diet for over fifty percent of the worldwide population. Of 503.117 MT of rice produced globally, with Bangladesh accounting for 7.0% of the total. Since gaining independence, the quantity produced of rice has risen three times, from 11 MT in 1971–1972 to 36.6 MT in 2019–20. With 11.55 million hectares of cultivated gross land, the nation achieves autarky to suit the needs of its 160.04 million citizens[1]. Sadly, it is one of the plants that is vulnerable to pests and diseases. Plant diseases, and how to identify and manage them to improve the health of crops, represent one of the most significant issues facing agriculture and have an impact on its trade. Plants experience various mechanical, morphological, and biochemical alterations when they become diseased [2]. Diseases can reduce output and cause environmental damage if they are not promptly diagnosed and treated. Finding the disease and identifying the type of disease as soon as possible is therefore crucial [3]. Currently, the majority of rice disease identification is done manually using biological methods in the field, which not only uses up a lot of time and resources but also has a low identification rate and no real-time efficacy, frequently missing the best window for disease control. In order to increase rice yields, cut expenses, and stop diseases from spreading further, it is crucial to apply

computer vision algorithms for the timely and accurate diagnosis of rice diseases as well as for automatic and efficient disease detection of rice. The majority of recent research on classifying and identifying plant diseases has been solely on the direction in which machine learning or deep learning algorithms should be applied. According to studies, image-processing methods are useful tools for identifying and classifying plant diseases [4]. The deep convolutional neural network (CNN) has been mostly utilized for image classification among the many deep learning approaches. The collecting of photos with rice disease is a difficult undertaking because CNN needs a lot of training data to be trained properly. The use of Transfer Learning can overcome this constraint. The transfer learning method updates the weights for the current small dataset using previously trained networks on large datasets [5]. Plant diseases and pests can be prevented and controlled with the aid of early diagnosis, which is also crucial for managing and making decisions regarding agricultural productivity.

1.2 Problem Statement

In the coming years, global rice production is expected to face increasing challenges, particularly in regions like Bangladesh, where rice security is synonymous with food security. To ensure rice security for future generations, it is imperative to address the persistent threat of diseases that significantly impact both the quality and quantity of rice yields [6]. In Bangladesh, various diseases, such as blast, brown spot, false smut, and others, pose significant risks, potentially resulting in yield losses ranging from 9.5% to 69% [7]. Monitoring these diseases, detecting their occurrences early, and implementing timely treatments are critical for minimizing losses and planning effective disease prevention strategies [8]. However, traditional methods of disease monitoring and management are often labor-intensive and time-consuming. In this context, machine learning (ML) systems offer promising solutions for automating disease detection and classification processes. Nonetheless, the accuracy of ML systems heavily relies on effective feature selection techniques, which can be challenging, especially when dealing with limited datasets. Obtaining large-sized datasets for training ML models for disease detection in agriculture poses a significant obstacle. To address these challenges, this study proposes the use of transfer learning, a technique that leverages pre-trained models on large datasets to enhance the

performance of models trained on smaller, domain-specific datasets. Specifically, three CNN modules, namely VGGNet, ResNet50, and InceptionResNetV2, are investigated for their effectiveness in detecting and classifying rice diseases. By exploring transfer learning and evaluating different CNN architectures, this study aims to develop robust and accurate models capable of detecting and classifying rice diseases efficiently, thus contributing to the preservation of rice security and agricultural sustainability in regions like Bangladesh.

1.3 Motivation for this Study

The prevalence of rice as a staple food source, coupled with the substantial global production of this cereal grain, underscores its critical importance for sustaining a significant portion of the world's population. Despite its significance, manual methods currently dominate the identification of rice diseases in fields, resulting in time and resource-intensive processes with limited efficacy and real-time capabilities. This inefficiency not only hampers disease identification rates but also compromises the timely implementation of disease control measures, thus exacerbating the spread of illnesses and impeding optimal crop yields. In response to these challenges, there is a pressing need to harness computer vision algorithms for the swift and accurate diagnosis of rice diseases, facilitating automated and efficient disease detection processes. While existing research has primarily focused on applying machine learning and deep learning algorithms for disease classification, image-processing techniques have emerged as valuable tools for disease identification and classification, particularly in plant pathology. Among these approaches, deep convolutional neural networks have garnered significant attention for their effectiveness in image classification tasks. However, the collection of sufficient training data, particularly images of rice diseases, poses a significant challenge due to the extensive data requirements of CNNs. To address this limitation, transfer learning presents a promising solution by leveraging pre-trained networks on large datasets to update weights for smaller, domain-specific datasets. By adopting transfer learning, this study aims to overcome the data scarcity hurdle and enhance the accuracy and efficiency of rice disease detection. Ultimately, the timely diagnosis of plant diseases, including those affecting rice crops, is imperative for effective disease management and informed decision-making to optimize agricultural productivity. This study is

motivated by the urgent need to develop robust and scalable solutions that empower farmers with the tools and insights necessary to combat plant diseases and safeguard global food security.

1.4 Research Objectives

Transfer learning is a pivotal methodology in deep learning, particularly renowned for its efficacy in image classification tasks. In this study, transfer learning is selected with specific objectives in mind:

1. Achieving Higher Recognition Accuracy of Rice Diseases: By leveraging transfer learning, the study aims to enhance the precision and efficiency of rice disease detection at an early stage. Utilizing pre-trained models allows for the extraction of intricate features from vast datasets, facilitating the accurate identification of various diseases afflicting rice crops.

2. Performance Comparison of VGGNet, ResNet50, and InceptionResNetV2: The study seeks to conduct a comparative analysis of three prominent convolutional neural network architectures: VGGNet, ResNet50, and InceptionResNetV2. Through this comparison, the performance nuances of each model can be elucidated, shedding light on their respective strengths and weaknesses in the context of rice disease classification. This analysis enables researchers to discern the most suitable model for the task at hand, thereby optimizing the overall effectiveness of the disease detection system.

1.5 Contribution of this Study

The major contributions of this thesis can be summarized as follows:

- i. Utilizing transfer learning to adapt pre-trained models over conventional CNN models for rice disease detection, enhancing the efficiency and accuracy of disease identification with a comparative analysis of optimizers used.

- ii. Demonstration of the transferability of these models to other crops underscores their versatility and potential for broader applications in agricultural contexts, offering insights into crop health management beyond rice cultivation.

1.6 Scope

This research focuses on the detection and classification of rice diseases in Bangladesh, specifically targeting common diseases such as blast, brown spot, and false smut. The study employs transfer learning techniques using three convolutional neural network (CNN) architectures: VGGNet16, ResNet50, and InceptionResNetV2, leveraging pre-trained models to improve detection accuracy. The dataset comprises rice leaf images sourced from agricultural research institutions and publicly available databases, with extensive preprocessing to ensure data quality. Performance evaluation is based on metrics including accuracy, precision, recall, and F1 score, with comparative analysis against previous methods. While the study is limited by the dataset size and computational resources, it provides valuable insights into the potential of transfer learning for agricultural applications. Future research may explore additional CNN architectures, larger datasets, and further optimization to enhance model performance. The findings have significant implications for improving rice disease management and can potentially be adapted for other crops and agricultural regions.

1.7 Research Outline

The rest of this report is organized as follows: Chapter 2 describes past and recent efforts made in rice disease detection using multiple machine learning and deep learning approaches. Chapter 3 elaborates on the measures taken for conducting this study. In Chapter 4, the expected result is discussed. Chapter 5 describes this study's conclusion, future scope, and limitations.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Detecting rice diseases early is crucial for implementing preventive measures and minimizing crop losses. Previous research in rice disease detection has focused on various techniques and methodologies to achieve accurate and timely identification of diseases. In recent years, advancements in computer vision, machine learning, and deep learning have revolutionized the field of rice disease detection. Numerous studies have explored the use of image processing techniques and artificial intelligence algorithms to automate and improve the efficiency of disease identification processes. These studies have leveraged datasets consisting of images of diseased rice plants, annotated with labels indicating the type of disease present. Early approaches to rice disease detection often involved handcrafted feature extraction methods, where domain-specific features were manually designed and extracted from images. These features were then used to train traditional machine learning classifiers, such as support vector machines (SVMs) or decision trees, for disease classification. While these methods showed promising results, they often lacked scalability and generalization capabilities across different types of diseases and environmental conditions. This chapter discusses the previous as well as recent works done in detecting rice diseases.

2.2 Objectives of the Review

The objective of the literature review is to critically examine and synthesize existing research and developments in the field of rice disease detection. Specifically, the literature review aims to

- i. Provide a comprehensive overview of the current state-of-the-art techniques, methodologies, and technologies employed in rice disease detection.
- ii. Evaluate the effectiveness of transfer learning as a technique for enhancing the accuracy, efficiency, and scalability of rice disease detection models.
- iii. Highlight emerging trends, advancements, and opportunities for future research in the field of rice disease detection, particularly in leveraging transfer learning and deep learning techniques to address existing challenges and limitations.

2.3 Types of Rice Diseases

Rice disease poses a significant challenge to rice cultivation in Bangladesh, a country heavily reliant on rice as a staple food crop and a vital component of its agricultural economy. Various diseases affect rice production in Bangladesh, leading to significant yield losses and economic impacts [9]. These diseases, among others, pose significant threats to rice production in Bangladesh, where agriculture serves as a primary source of livelihood for millions of people [10]. The humid climate, conducive to disease development, coupled with intensive rice cultivation practices, exacerbates the spread and severity of rice diseases in the country. Some of the prominent rice diseases in Bangladesh are included in Table 2.1:

Table 2.1 Types of Rice Diseases

Name	Origin	Affected Area	Yield Loss	Reference
------	--------	---------------	------------	-----------

Blast	<i>magnaporthe oryzae</i>	Leaves, nodes, necks, and panicles	33.7 to 98%	[11]
Brown Spot	<i>bipolaris oryzae</i>	Small, dark-brown lesions with yellow halos	45%	
Sheath Blight	<i>rhizoctonia solani</i>	Lesions on the leaf sheaths and panicles	14 to 31%	[12]
Bacterial Leaf Blight	<i>xanthomonas oryzae</i> pv. <i>Oryzae</i>	Water-soaked lesions on rice leaves	4.8 to 30.4%	[11]
Tungro	Rice tungro virus transmitted by insect vectors	Yellowing of leaves, and reduced tillering	100% in severe cases	[13]
Stem rot	<i>sclerotium oryzae</i>	Formation of white, fluffy mycelial growth on the stems	35%	[11]
Sheath rot	<i>sarocladium oryzae</i>	Brown lesions with black margins in the sheaths of rice plants	14 to 31%	[12]
False smut	<i>ustilaginoidia virens</i>	Growth of velvety spores that enclose floral parts	87%	[11]

2.4 Distinct Neural Networks for Rice Disease Prediction

Distinct CNN Techniques in image classification refer to the utilization of various CNN architectures tailored for the task of classifying images into different categories. These techniques leverage the deep learning capabilities of CNNs to automatically learn and extract hierarchical features from input images, enabling accurate and efficient classification. The Visual Geometry Group Network (VGGNet) is a CNN architecture developed by the Visual Geometry Group at the University of Oxford. It is characterized by its simplicity and uniform architecture, consisting of multiple convolutional layers with small 3x3 filters and max-pooling layers. VGGNet variants, such as VGG-16 and VGG-19, have been widely used for image classification tasks [14]. Residual Networks (ResNet) are a family of CNN architectures introduced by Microsoft Research. They feature skip connections or residual connections that enable the training of very deep networks without vanishing gradient problems. ResNet variants, such as ResNet-50 and ResNet-101, have achieved state-of-the-art performance on various image classification benchmarks [14]. GoogLeNet, also known as Inception, is a CNN architecture developed by Google researchers. It introduced the concept of inception modules, which utilize parallel convolutional operations of different filter sizes to capture multi-scale features effectively [15]. Dense Convolutional Networks (DenseNet) are CNN architectures that establish dense connections between layers, where each layer receives direct input from all preceding layers. DenseNet encourages feature reuse and facilitates gradient flow, resulting in improved feature learning and model performance [16]. Researchers often design customized CNN architectures tailored for specific image classification tasks. These architectures may incorporate novel design elements, such as attention mechanisms, capsule networks, or spatial transformers, to enhance feature learning and classification accuracy. In the task of rice disease detection, transfer learning holds significant relevance and benefits. Transfer learning allows us to utilize pre-trained models trained on large-scale datasets, such as ImageNet, which contain millions of labeled images across diverse categories. By transferring knowledge from these datasets to the rice disease detection task, we can effectively leverage the rich representations learned by the pre-trained models. By fine-tuning the pre-trained models on rice disease images, we can adapt the learned features to the specific characteristics of rice plants and diseases, thereby improving the model's ability to discriminate between healthy and diseased plants.

The proposed model by Shreya et al. gathered dataset from the internet and rice fields is used to train and test the VGG-16 model. accuracy is 92.46% [17]. ResNet50, ResNet101, DenseNet161, and DenseNet169 models were employed by Mathulapransan et al. to compare various models [8]. CNN is trained using 87,848 photos in the P. Konstantinos et al. publication, with 25 plant kinds divided into 58 classes, including healthy plants. Different types of models were trained, and the top model had an identification accuracy of 99.53% [18]. Ashiqul et al. examined four models, including VGGNet-19, InceptionResNet-V2, ResNet-101, and Xception, and found that InceptionResNet-V2 performed better [19]. Junde Chen et al. examined VGGNet pre-trained on ImageNet and the Inception module and the weights are initialized using the pre-trained networks on ImageNet's sizable labeled dataset. On the public dataset, it has a validation accuracy of 91.83% [20]. The DenseNet pre-trained on ImageNet and Inception module was chosen to be employed in Junde Chen et al., combining the benefits of both, and this approach exhibits superior performance in comparison to other state-of-the-art methods. In the publicly available dataset, it obtains an average prediction accuracy of at least 93.07% [21]. A high-accuracy transfer learning model that can offer farmers a mobile option was proposed by Kathiresan et al. To balance the quantity of disease samples, this study also uses a generative advertisement network. The average cross-validation for this model is 98.79% [22]. Sethy et al. took a more thorough method, using an SVM classifier, deep feature extraction, and 11 transfer learning networks [23]. Sharma et al. suggested a customized CNN that divided rice leaf diseases into three classes using innovative architectures [24]. A customized feature extraction architecture was proposed by Liang et al. and assessed using an SVM classifier along the same lines. The authors looked into a method for categorizing the five diseases that affect banana leaves. Using a variety of picture pre-processing filters, they deleted the image's undesirable information during the pre-processing stage. To remove the diseased region of the leaf from the pre-processed image for segmentation, they used genetic algorithms. From the segmented image, scientists later extracted color and texture information. Finally, the SVM classifier receives these extracted characteristics as input, and the classification is carried out with an accuracy of 94.7% [25]. To categorize diseases of various plant leaves, the authors presented a CNN model known as depth-wise separable. In comparison to VGGNet, they employed MobileNet to achieve a classification accuracy of 98.34% while using fewer parameters [26]. To identify 59 illnesses affecting various crops, researchers created the unique model MDFC-ResNet (Multi-Dimensional Feature Compensation Residual Neural Network). This technique can

notify farmers of symptoms of plant diseases and also identify plant diseases automatically. Using the Keras framework, they ran experiments and got improved accuracy - 84.22% [27]. In order to classify four diseases of the grape leaf in a real-time setting, the authors suggested an improved DCNN. They have employed 4449 grape leaf photos throughout the investigation. Better results were obtained using the Inception-ResNet-v2 module, which had a mean average precision (mAP) of 81.1% [26]. A LINE Bot System was created by Pitchayagan et al. to identify rice illnesses from photos captured in a paddy field using deep learning techniques. The deployment model's performance increased from 91.1% to 94.6% [28]. Three learning algorithms for the categorization of plant disease were used by the authors to test different CNN architectures using plant village datasets. A 99.76% accuracy rate was achieved with the Inceptionv3 model [29]. To classify nine tomato leaf diseases, the authors trained two architectures (GoogLeNet and AlexNet). Later, the model was used to classify various tomato plant leaf diseases with 99.18% accuracy by applying the characteristics to the classification layer [30]. Table 2.2 is a summary of the previous works on rice disease detection and dissimilarities with this study

Table 2.2 Related works and their dissimilarities with this study

Reference	Dataset Classes	Dataset Source	Amount of Data	Method used	Result	Dissimilarities
[8]	Five diseases classes	Thailand	Large dataset of 12,223 images	ResNet 50, ResNet101, DenseNet 161, DenseNet169	94.74% on DenseNet	Do not use VGGNet, or InceptionResNetV2 module.
[17]	Three disease classes	West Bengal, India	1,649	VGGNet -16	92.46%	Do not use ResNet, or InceptionResNetV2 module.
[18]	25 different plants	An open database	87,848	AlexNet, AlexNetOWTBN, GoogLeNet, Overfeat, VGGNet	99.53%	Do not use ResNet50 Not implemented on rice plant

[19]	4 classes of paddy diseases	Cultivation field in Bangladesh	984	VGGNet-19, InceptionResNet, ResNet101, XceptionNet	92.68%	Do not use the same models. Different diseases were identified.
[20]	8 classes of rice and maize diseases	Cultivation field in China	966	VGGNet, Inception, ResNet50, Hybrid INC-VGG	92%	Uses hybrid model. Different diseases were identified.
[21]	5 classes of rice diseases	Cultivation field in China	500	VGGNet, Inception, DENS-INCEP	92.86%	Uses hybrid model.
[22]	8 classes of rice and leaves diseases	Three open sources of datasets	10,358	MobileNet, Inception, ResNet, Xception, RiceDenseNet	98.79%	Utilizes Generative Adversarial Network (GAN) augmented data. Proposes a new model.
[23]	4 classes of rice leaf diseases	Odisha, India	5,932	11 deep CNN models with SVM classifier	98.38%	SVM Classifier used.
[24]	3 classes of rice diseases	Previous dataset	Not specified	VGGNet-16	91%	Do not use ResNet, or InceptionResNetV2 module.
[27]	49 detailed diseases	AI Challenger	36,258	MDFC ResNet	84.22%	Uses optimization model Not implemented on rice plants
[28]	6 classes of rice diseases	Not specified	6,538	YOLOV3	94.6%	Uses a completely different model and proposes real time implementation

[31]	4 classes of rice diseases	Internet	5932	RiceDRA-Net	99.71%	Implements a new deep neural network and constructs a new dataset
[32]	3 classes of rice diseases	Online resources	400	SVM classifier	93.16%	Analysis of classifiers in disease detection

2.5 Different Convolution Neural Networks and Their Scope

The scope of CNNs in agriculture is vast, ranging from early-stage plant disease detection to yield prediction and agricultural automation. By leveraging the capabilities of different CNN architectures, researchers and practitioners can develop sophisticated AI solutions to address the evolving challenges in modern agriculture. Different CNNs offer unique capabilities and scopes in agriculture, contributing to various tasks such as crop monitoring, disease detection, yield prediction, and agricultural automation. VGGNet is widely used for image classification tasks in agriculture, including plant species recognition, leaf disease detection, and crop yield estimation. Its simple architecture with small convolutional filters makes it effective for feature extraction from agricultural images. ResNet's ability to train very deep neural networks is beneficial for complex agricultural tasks requiring hierarchical feature learning. It is employed in leaf disease detection, crop pest identification, and yield prediction models, where capturing intricate features from high-resolution images is crucial. InceptionNet's inception modules enable efficient multi-scale feature extraction, making it suitable for tasks like plant disease diagnosis, fruit detection, and plant phenotyping. Its parallel convolutional operations facilitate the detection of fine-grained details in agricultural images. DenseNet's densely connected layers encourage feature reuse and facilitate gradient flow, enhancing feature learning in agricultural images. It finds applications in plant disease classification, crop yield estimation, and weed detection, where capturing complex relationships between image features is crucial. Customized CNN architectures tailored for specific agricultural tasks leverage innovative design elements to address domain-specific

challenges. These architectures incorporate attention mechanisms, spatial transformers, and capsule networks to improve the accuracy and robustness of agricultural models.

While CNNs have demonstrated remarkable success in various tasks, they are not without limitations. CNNs are often described as "black box" models because it can be challenging to interpret the decisions they make. Understanding how and why a CNN arrives at a particular classification can be difficult, especially for complex networks with many layers. CNNs typically require large amounts of labeled training data to achieve high performance. Obtaining and annotating such datasets can be time-consuming and costly, especially for niche or specialized domains. Training deep CNNs may require powerful hardware such as GPUs or TPUs and substantial computational resources. The transfer learning technique has been widely used in recent works to train classification models. It is a powerful technique in deep learning that addresses several limitations of CNN models. Transfer learning allows CNN models to leverage pre-trained networks trained on large datasets, such as ImageNet. By transferring knowledge from these datasets, models can achieve high performance with smaller, domain-specific datasets, reducing the need for extensive labeled data collection. Pre-trained CNN models already contain learned features from large datasets, which can accelerate the training process for domain-specific tasks. This results in faster convergence and reduced computational resources required for training.

2.6 Research Gap

Based on the literature provided, several research gaps can be identified in the field of rice disease detection

- i. **Limited Exploration of Model Architectures:** While some studies have explored multiple model architectures such as VGG-16, ResNet, DenseNet, InceptionResNet-V2, and Xception, there may be other architectures that have not been thoroughly investigated.
- ii. **Lack of Consistency in Evaluation Metrics:** There appears to be inconsistency in the evaluation metrics used across different studies, with some focusing solely on accuracy while others consider additional metrics like precision, recall, and F1-score.

- iii. **Limited Diversity in Datasets:** Many studies rely on datasets gathered from the internet or specific rice fields, which may lack diversity in terms of environmental conditions, rice varieties, and disease types.
- iv. **Insufficient Exploration of Data Augmentation Techniques:** There appears to be limited exploration of advanced data augmentation techniques in some studies. Investigating and optimizing data augmentation strategies could lead to better performance in rice disease detection models.
- v. **Limited Focus on Real-time Detection:** While some studies mention the deployment of models for real-time disease detection in agricultural settings, there is a lack of detailed exploration and evaluation of the real-time performance of these models.

2.7 Summary

This chapter has provided a comprehensive overview of the advancements and methodologies in rice disease detection, emphasizing the critical importance of early detection to mitigate crop losses. Traditional methods, while initially effective, often lack the scalability and generalization required for diverse disease conditions and environments. The advent of computer vision, machine learning, and deep learning techniques has revolutionized the field, enabling more efficient and accurate disease identification. Despite the advancements, several research gaps including the limited exploration of diverse CNN architectures, inconsistency in evaluation metrics, lack of dataset diversity, insufficient utilization of advanced data augmentation techniques, and a limited focus on real-time detection capabilities. Addressing these gaps can lead to more robust, scalable, and practical solutions for rice disease detection, ultimately contributing to improved agricultural productivity and sustainability.

Chapter 3

METHODOLOGY

3.1 Introduction

The methodology section of this thesis elucidates the systematic framework employed to investigate the detection of rice diseases through transfer learning, leveraging CNN architectures including VGGNet16, InceptionResNetV2, and ResNet50. In the context of Bangladesh's agrarian landscape, where rice cultivation dominates, timely and accurate disease detection is paramount for safeguarding agricultural productivity and ensuring food security. This section outlines the steps undertaken to acquire, preprocess, and train models on rice disease datasets, model architectures, justification for selecting the models, and the testing process. By delineating the methodological approach in detail, this study aims to contribute to the advancement of automated disease detection systems in rice farming, facilitating early intervention and enhancing overall agricultural productivity.

3.2 Model Development

In this study, three prominent CNN architectures—VGGNet16, ResNet50, and InceptionResNetV2—are harnessed for the task of rice disease detection using a transfer learning approach. Each of these models brings its unique architectural features and depth, allowing for

comprehensive exploration of the dataset and effective learning of discriminative features crucial for disease classification. Transfer learning, a technique widely adopted in deep learning, forms the cornerstone of this research. Leveraging pre-trained versions of these CNNs, initially trained on large-scale image datasets such as ImageNet, it was possible to capitalize on the knowledge gained from diverse visual recognition tasks. This knowledge is then fine-tuned and adapted to the specific nuances of rice disease detection, enabling the models to effectively discern subtle patterns indicative of various rice diseases. During the transfer learning process, the pre-trained CNN models are reconfigured to accommodate the intricacies of the rice disease dataset. The final layers of these architectures are replaced to suit the classification task at hand, while the earlier layers retain their learned feature extraction capabilities. This strategic adaptation allows for efficient model training on relatively smaller datasets, mitigating the risk of overfitting and enhancing the generalization ability of the models. Post-training, the adapted CNN models undergo rigorous evaluation using accuracy as a performance metric. The metric provides a comparison between the four models utilized in this research, thereby facilitating informed decision-making for farmers and agricultural stakeholders.

3.2.1 Neural Network Models

Neural network models are typically used for solving complex problems in machine learning and artificial intelligence, including pattern recognition, classification, regression, clustering, and sequence generation. These models learn by adjusting the weights and biases of connections between neurons during a process called training. Training involves presenting input data to the network, comparing the predicted outputs with the true outputs, and updating the model parameters using optimization algorithms, such as gradient descent, to minimize the prediction error. The four neural networks, conventional and customized models, used in this study are as follows:

a) Convolution Neural Network

Convolutional Neural Networks are separate from other neural networks because of their superior performances with image, speak, or audio signal entries. Comprising three key types of layers—

convolutional, pooling, and fully-connected (FC), CNNs are adept at capturing intricate patterns and features within data. At the heart of a CNN, lies the convolutional layer, where the bulk of computation occurs. This layer operates by convolving input data with filters, also known as kernels, to generate feature maps. Following convolution, a Rectified Linear Unit (ReLU) transformation is typically applied to introduce nonlinearity, enhancing the model's representational power. Pooling layers, on the other hand, facilitate dimensionality reduction by down sampling the feature maps. The final layer of a CNN, the fully-connected layer, orchestrates the classification task based on extracted features from preceding layers. While convolutional and pooling layers often employ ReLU activation functions, fully-connected layers typically utilize SoftMax activation for probabilistic classification. Figure 3.1 shows the model architecture of the CNN model used in this study

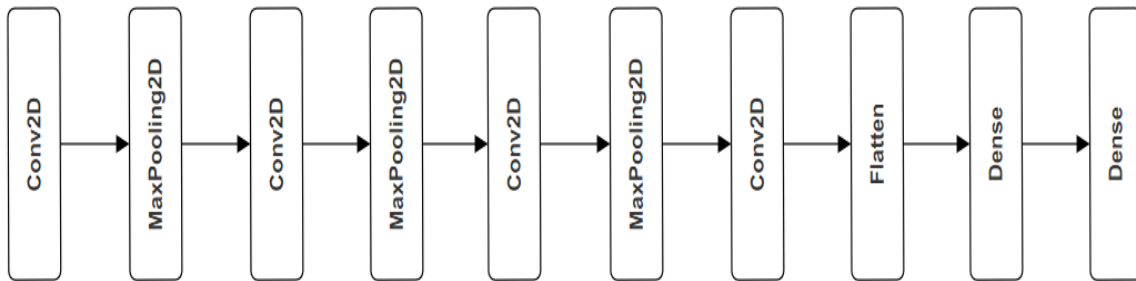


Figure 3.1 CNN model architecture

b) VGGNet16

The key distinguishing feature of VGG16 is its uniform architecture, where each convolutional layer consists of 3x3 filters with a stride of 1 and same padding, followed by max-pooling layers with 2x2 filters and a stride of 2. This consistent design simplifies the architecture and makes it easier to understand and implement. The depth of VGG16 allows it to capture increasingly complex features as the data flows through the network. The initial layers learn basic features such as edges and textures, while deeper layers learn higher-level features and abstract representations of objects present in the input images. This hierarchical feature extraction enables VGG16 to

achieve remarkable performance in image classification tasks, even on datasets with diverse and complex images. The model architecture for vgg16 trained for this study is shown in Figure 3.2

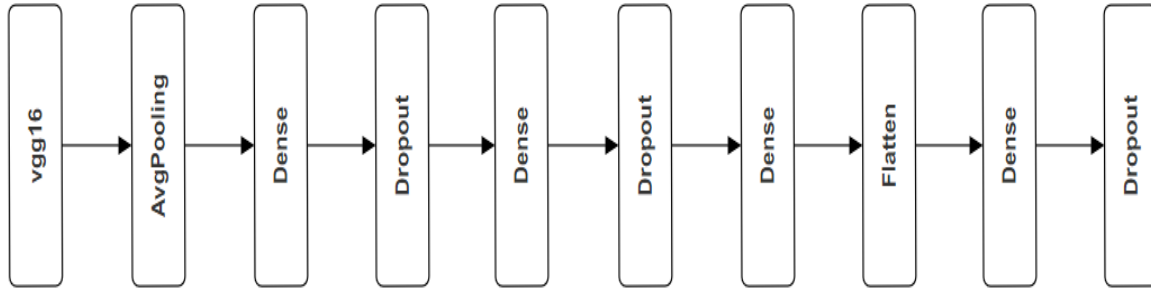


Figure 3.2 VGGNet 16 model architecture

c) ResNet50

ResNet50 comprises a stack of residual blocks, each containing multiple convolutional layers with batch normalization and ReLU activation functions. Additionally, the architecture includes down sampling layers, such as strided convolution or pooling layers, to reduce the spatial dimensions of the feature maps while increasing the number of filters. Figure 3.3 illustrates the model architecture of the resnet50 model used in this study.

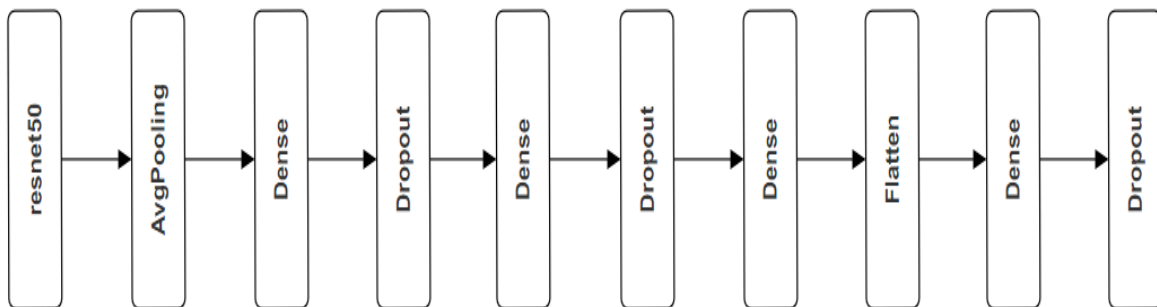


Figure 3.3 ResNet50 model architecture

d) InceptionResNetV2

InceptionResNetV2 is a sophisticated convolutional neural network architecture that combines the principles of both the Inception module and residual connections from ResNet. The architecture of InceptionResNetV2 is characterized by its intricate design, which features multiple modules of varying kernel sizes and depths within each inception block. These modules enable the network to capture features at different scales and resolutions, facilitating the extraction of rich and diverse features from input images. In addition, the residual connections allow for the direct propagation of gradients through the network, mitigating the vanishing gradient problem and enabling the training of deeper architectures. The model architecture used for this study is illustrated in Figure 3.4

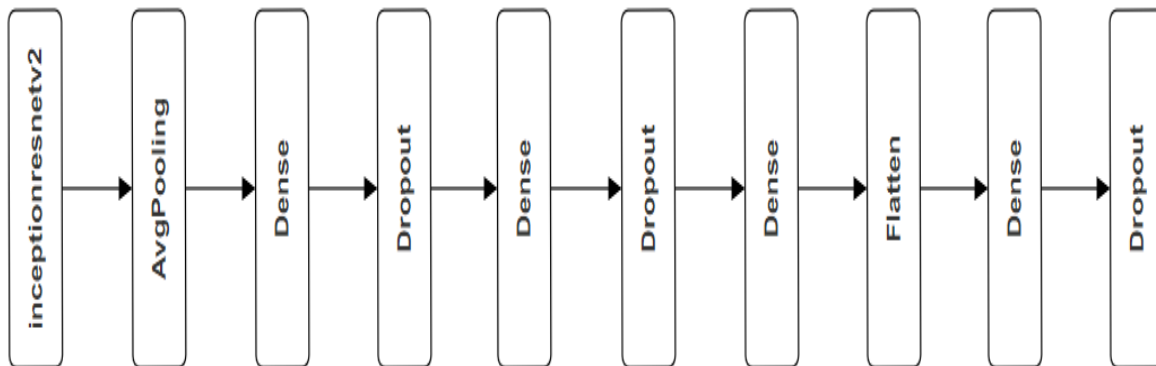


Figure 3.4 InceptionResNetV2 model architecture

3.2.2 Model Architecture

A model architecture summary provides a comprehensive overview of the structure and properties of a neural network model. It typically includes details such as the type and order of layers, the number of parameters, the output shape of each layer, and the total number of trainable parameters in the model.

a) CNN Architecture

As data traverses through the layers of a CNN, from simple features like colors and edges in early layers to more complex shapes and objects in later ones, the network progressively discerns finer details and nuances. This hierarchical approach allows CNNs to effectively identify and classify diverse inputs, contributing to their widespread utility in tasks ranging from image recognition to natural language processing. Table 3.1 gives a summary of the CNN model used

Table 3.1 Model Architecture of CNN

Layer (type)	Output Shape	Param #
conv2d_30 (Conv2D)	(None, 298, 298, 32)	896
max_pooling2d_30 (MaxPooling2D)	(None, 149, 149, 32)	0
conv2d_31 (Conv2D)	(None, 147, 147, 64)	18496
max_pooling2d_31 (MaxPooling2D)	(None, 73, 73, 64)	0
conv2d_32 (Conv2D)	(None, 71, 71, 64)	36928
max_pooling2d_32 (MaxPooling2D)	(None, 35, 35, 64)	0
conv2d_33 (Conv2D)	(None, 33, 33, 64)	36928
max_pooling2d_33 (MaxPooling2D)	(None, 16, 16, 64)	0
conv2d_34 (Conv2D)	(None, 14, 14, 64)	36928
max_pooling2d_34 (MaxPooling2D)	(None, 7, 7, 64)	0
conv2d_35 (Conv2D)	(None, 5, 5, 64)	36928
max_pooling2d_35 (MaxPooling2D)	(None, 2, 2, 64)	0
flatten_5 (Flatten)	(None, 256)	0
dense_10 (Dense)	(None, 64)	16448
dense_11 (Dense)	(None, 3)	195
Total params: 183747 (717.76 KB)		
Trainable params: 183747 (717.76 KB)		
Non-trainable params: 0 (0.00 Byte)		

b) VGGNet16 Architecture

The VGG16 model is a functional architecture comprising various layers, including convolutional and dense layers, designed for image classification tasks. The summary of this architecture is shown in Table 3.2

Table 3.2 Model Architecture of VGGNet16

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 9, 9, 512)	14714688
global_average_pooling2d (GlobalAveragePooling2D)	(None, 512)	0
dense (Dense)	(None, 512)	262656
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 256)	131328
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 128)	32896
dropout_2 (Dropout)	(None, 128)	0
flatten (Flatten)	(None, 128)	0
dense_3 (Dense)	(None, 64)	8256
dropout_3 (Dropout)	(None, 64)	0
dense_4 (Dense)	(None, 3)	195
=====		
Total params: 15150019 (57.79 MB)		
Trainable params: 435331 (1.66 MB)		
Non-trainable params: 14714688 (56.13 MB)		

The model begins with convolutional layers, which progressively extract features from the input images, resulting in a feature map with dimensions (None, 7, 7, 512). Subsequently, a global average pooling layer reduces the spatial dimensions of the feature map to (None, 512), aggregating the features across each channel. Following this, several dense layers are employed to further process the features, gradually reducing the dimensionality of the data. Dropout layers are

inserted after each dense layer to prevent overfitting by randomly deactivating a fraction of neurons during training. Finally, the model concludes with a dense output layer with three units, representing the number of classes in the classification task.

c) ResNet50 Architecture

This sequential model architecture begins with an input layer with dimensions (None, 224, 224, 3), suitable for processing RGB images. The ResNet50 model is then incorporated, which extracts features from the input images and produces a feature map with dimensions (None, 7, 7, 2048). Following this, a global average pooling layer aggregates the features across each channel, resulting in a feature vector of size (None, 2048). Subsequently, several dense layers are employed to further process the features, gradually reducing the dimensionality of the data. Dropout layers are included after each dense layer to prevent overfitting by randomly deactivating a fraction of neurons during training. The model concludes with a dense output layer containing three units, representing the number of classes in the classification task. This is demonstrated in Table 3.3

d) InceptionResNetV2 Architecture

The presented architecture in Table 3. utilizes the InceptionResNetV2 model, a pre-trained convolutional neural network, as its core feature extractor. The input layer accepts images with dimensions of 224x224 pixels and three color channels. The extracted features pass through the InceptionResNetV2 layers, resulting in feature maps with dimensions of 5x5x1536. A global average pooling layer then condenses these feature maps into a single vector, facilitating further processing. Subsequent dense layers, each followed by dropout layers, perform feature extraction and representation learning, gradually reducing the dimensionality of the data. Finally, a softmax-activated dense layer serves as the output layer, generating class probabilities for the classification task. This architecture combines the power of transfer learning from the InceptionResNetV2 model with additional dense layers for effective feature extraction and classification. Regularization techniques such as dropout layers help prevent overfitting, ensuring robust performance during training and inference. Table 3.4 gives a comprehensive summary of this architecture

Table 3.3 Model Architecture of ResNet50

Layer (type)	Output Shape	Param #
resnet50 (Functional)	(None, 10, 10, 2048)	23587712
global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 2048)	0
dense_5 (Dense)	(None, 512)	1049088
dropout_4 (Dropout)	(None, 512)	0
dense_6 (Dense)	(None, 256)	131328
dropout_5 (Dropout)	(None, 256)	0
dense_7 (Dense)	(None, 128)	32896
dropout_6 (Dropout)	(None, 128)	0
flatten_1 (Flatten)	(None, 128)	0
dense_8 (Dense)	(None, 64)	8256
dropout_7 (Dropout)	(None, 64)	0
dense_9 (Dense)	(None, 3)	195
=====		
Total params: 24809475 (94.64 MB)		
Trainable params: 1221763 (4.66 MB)		
Non-trainable params: 23587712 (89.98 MB)		

3.3 Justification for Model Selection

The selection of CNN architectures, namely VGGNet16, InceptionResNetV2, and ResNet50, for rice disease detection, is informed by their architectural characteristics and performance in similar tasks

i) VGGNet16 has demonstrated strong performance in various image classification tasks, including large-scale datasets like ImageNet. The straightforward design of VGGNet16 makes it

Table 3.4 Model Architecture of InceptionResNetV2

Layer (type)	Output Shape	Param #
inception_resnet_v2 (Functional)	(None, 8, 8, 1536)	54336736
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1536)	0
dense (Dense)	(None, 512)	786944
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 256)	131328
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 128)	32896
dropout_2 (Dropout)	(None, 128)	0
flatten (Flatten)	(None, 128)	0
dense_3 (Dense)	(None, 64)	8256
dropout_3 (Dropout)	(None, 64)	0
dense_4 (Dense)	(None, 3)	195
=====		
Total params: 55296355 (210.94 MB)		
Trainable params: 959619 (3.66 MB)		
Non-trainable params: 54336736 (207.28 MB)		

easy to understand and implement. Its deep architecture allows for the extraction of hierarchical features, which can be beneficial for capturing complex patterns associated with rice diseases. Moreover, VGGNet16's simplicity makes it a suitable candidate for transfer learning, enabling efficient adaptation to the rice disease detection task.

ii) ResNet50 has been widely adopted in various computer vision tasks, thanks to its ability to train deep networks effectively. By alleviating the vanishing gradient problem, ResNet50 enables the training of deeper architectures, which can capture increasingly complex features relevant to rice

disease detection. Its proven performance in image classification tasks makes it a reliable choice for transfer learning, offering robustness and efficiency in adapting to the nuances of rice disease classification.

ii) InceptionResNetV2 excels in capturing diverse and fine-grained features from high-resolution images, making it well-suited for complex classification tasks. Its utilization of residual connections mitigates the challenges associated with training deep networks, enabling effective learning of rich representations. This architectural versatility and depth make InceptionResNetV2 a compelling choice for rice disease detection, particularly when dealing with diverse and subtle disease symptoms.

3.4 Transfer Learning Framework

Transfer learning is a machine learning technique where a model trained on one task is repurposed or fine-tuned for a related task. Rather than training a model from scratch on a new dataset, transfer learning leverages knowledge gained from previous training to improve performance on the target task. This approach is particularly useful when the target dataset is small or lacks sufficient labeled data for training. The process of transfer learning typically involves two stages:

1) Pre-training Stage: In this stage, a deep learning model is trained on a large-scale dataset, often referred to as the source dataset. The model learns to extract general features and patterns from the input data, which are relevant to various tasks. Popular pre-trained models are often trained on datasets like ImageNet, which contains millions of labeled images across thousands of categories.

2) Fine-tuning Stage: After pre-training on the source dataset, the pre-trained model is adapted or fine-tuned to the target task using a smaller dataset, known as the target dataset. The parameters of the pre-trained model are adjusted or fine-tuned during training on the target dataset to adapt the learned features to the specific characteristics of the target task.

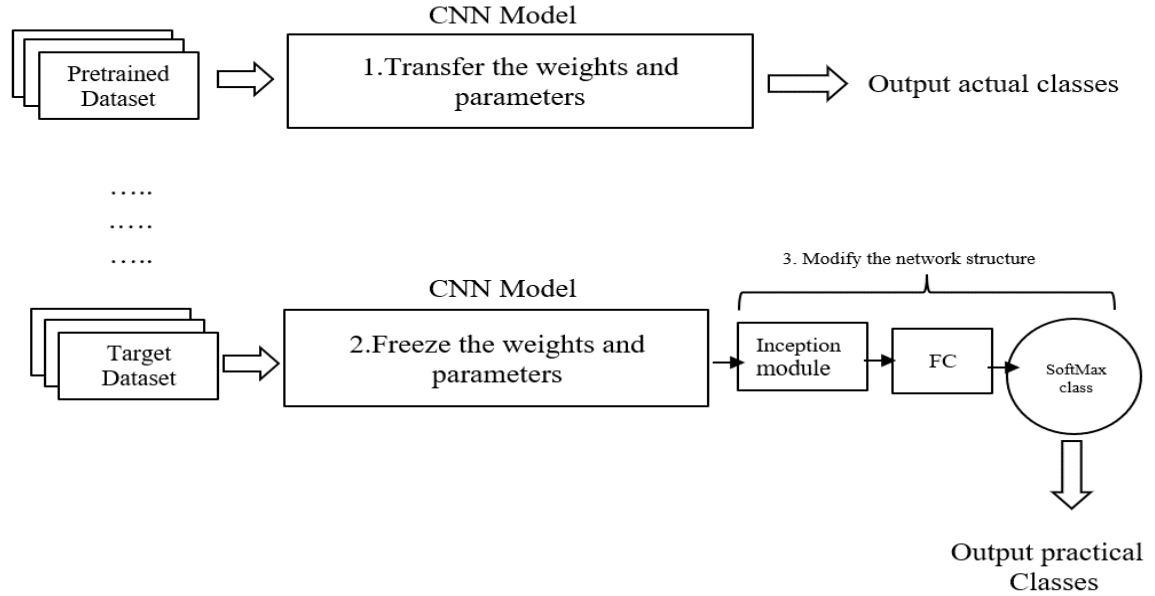


Figure 3.5 Transform Learning Framework

3.5 Workflow

This section focuses on how the work will be implemented. Figure 3.6 illustrates the workflow that will be maintained for conducting this research. This study will start with collecting data followed by image pre-processing – resize and rescaling, pretrained CNN models, Transfer Learning for improving generalization, and performance analysis. Once the models are trained, tested, and validated, the model with the best performance metric will be selected as the best CNN model for rice disease detection.

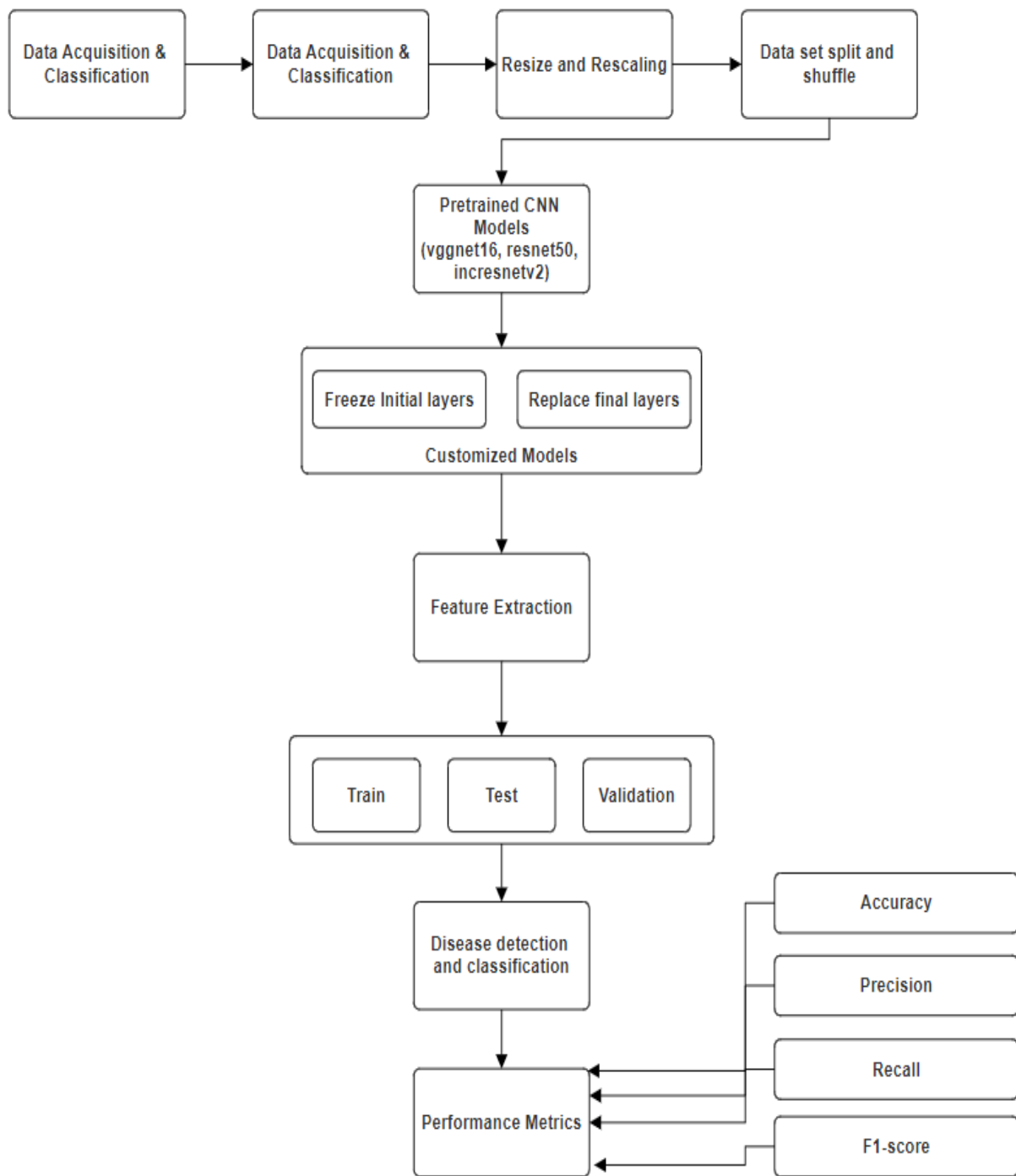


Figure 3.6 Overview of the Workflow

3.6 Dataset Preparation

By meticulously preparing the dataset, the rice disease detection model can effectively learn and discern patterns associated with different types of rice diseases.

3.6.1 Dataset Acquisition

In the pursuit of developing an effective rice disease detection model, a fundamental aspect lies in the acquisition and preparation of high-quality datasets. Leveraging internet sources, a comprehensive dataset comprising 4,330 images of infected rice leaves has been meticulously curated. This dataset encompasses instances of three prevalent rice diseases: blast, bacterial blight, and tungro, each posing significant threats to rice production globally. Figure 3.7 shows each image from each three classes.

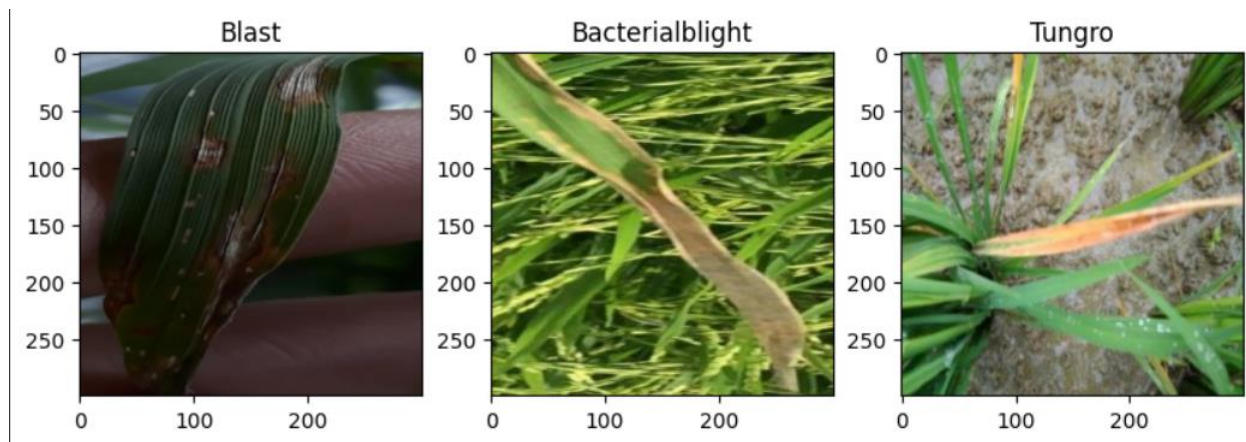


Figure 3.7 Sample images of rice leaf dataset

3.6.2 Image Resize & Rescaling

Once classified, all the images are resized to a uniform size of 224x224 pixels to match the models. The length of datasets is determined by the number of batches. The dataset has been meticulously labeled and classified into distinct categories, ensuring clarity and precision in identifying the types of rice diseases represented in each image. Additionally, the pixel values of the images are rescaled using the rescaling transformation function where the rescaling factor is $1/255$. This operation ensures that the pixel values of the image are divided by 255 so that pixel values are normalized to the range $[0, 1]$.

3.7 Training and Testing the Models

This section outlines the systematic approach employed to train and test the neural models for rice disease detection. The training procedure for all four models - CNN, VGG16, ResNet50, and InceptionResNetV2 follows a standardized approach, where the only variation lies in their respective architectures. Each model is compiled with the same optimization algorithms (Adam, SGD, RMSprop). During the testing phase, the models are tested using the test dataset and evaluate the model's performance to make a comparative analysis.

3.7.1 Training and Testing Dataset Split

The meticulous division of the dataset enables robust model training while ensuring unbiased evaluation of model performance on unseen data. By systematically allocating data for training, validation, and testing, the efficacy and generalization ability of the trained models can be rigorously assessed, ultimately guiding the selection of the most suitable model architecture for rice disease detection. Following established conventions, the dataset has been divided into a train-test split with a ratio of 70:30, where 70% of the data is allocated for training and 30% for testing. Subsequently, the testing subset is further subdivided into testing and validation sets with a split ratio of 50:50. This ensures that 10% of the original dataset is reserved for validation purposes during the training phase after each epoch, aiding in monitoring model performance and preventing

overfitting. The remaining 10% of the dataset is exclusively designated for measuring the accuracy of the models, facilitating comparative analysis to identify the most effective model architecture.

Table 3.5 Train Split Ratio

Class	Total images	Training	Testing	Validation
3	4330	3030	650	650

3.7.2 Training Phase

The training procedure begins by defining the input shape of the images, and the number of classes. Subsequently, a Sequential model is constructed using the Keras Sequential API. The models were trained using a batch size of 64. Following 30 epochs of training, the models exhibited commendable performance on the datasets. The training data is fed into each model for a fixed number of epochs, with the same batch size used for each iteration. During training, the models learn to extract features and patterns from the input images through their respective architectures. Through iterative training and validation processes, the model refines its ability to accurately identify and classify rice diseases, thereby empowering farmers and agricultural stakeholders with timely insights for disease management and mitigation strategies. The optimization algorithm updates the model's parameters based on the computed loss, gradually improving performance over successive epochs.

3.7.3 Optimization

Neural networks require optimization algorithms to update their weights and biases during training, and without an optimizer, the model won't learn effectively. During the training phase of the neural models, three distinct optimizers are employed: 'adam', 'sgd' (Stochastic Gradient

Descent), and 'rmsprop' (Root Mean Square Propagation). These optimizers serve as pivotal components in the training process, responsible for iteratively adjusting the model's parameters to optimize its performance [33].

The 'adam' optimizer combines the advantages of both momentum optimization. It maintains separate learning rates for each weight, dynamically adapting them based on the first and second moments of the gradients. This adaptive nature allows 'adam' to efficiently handle sparse gradients and noisy data, making it well-suited for a wide range of optimization tasks.

The 'sgd' optimizer is a classic optimization algorithm widely used in machine learning. It updates the model's parameters by computing the gradient of the loss function concerning each parameter and adjusting them in the direction that minimizes the loss. While 'sgd' is straightforward to implement, it can struggle with noisy or sparse gradients and may converge slowly compared to more advanced optimizers.

Lastly, the 'rmsprop' optimizer addresses some of the limitations of 'sgd' by adapting the learning rate for each weight based on the magnitude of recent gradients. By dividing the learning rate by the root mean square of the exponentially decaying average of squared gradients, 'rmsprop' effectively scales the learning rates according to the relevance of each parameter. This adaptive learning rate mechanism enables 'rmsprop' to converge faster and more reliably, particularly in scenarios with non-stationary objectives or sparse gradients.

3.7.4 Testing Phase

To assess the model's performance on the test dataset, each model's performance are tested against the test dataset, which returns the loss value and accuracy score. Additionally, a `prediction()` function is defined to make predictions on individual images. Inside this function, the image is converted to a NumPy array, reshaped to create a batch, and then passed through the model to obtain predictions. The predicted class and confidence level are determined based on the highest probability in the prediction output.

3.8 Measurement Factors

Metrics for classification that will be used for the performance analysis for the proposed models are accuracy, precision, recall, and F1-score. Before diving into these, let us comprehend True Positive, True Negatives, False Positive, False Negatives, accuracy, precision, F-measure, specificity, and recall. These terms are defined as follows:

True Positive (*TP*): True positives are cases when the actual classes of the data point are True and the predicted is also True.

True Negatives (*TN*): True Negatives are cases when the actual classes of the data point are False and the predicted is also False.

False Positive (*FP*): False positives are cases when the actual classes of the data point are False and the predicted is also True.

False Negatives (*FN*): True positives are cases when the actual classes of the data point are True and the predicted is also False.

i. **Confusion Matrix**

A confusion matrix is a structured table commonly employed to evaluate the effectiveness of a classification model using a dataset of known true values. It allows visualization of the performance of an algorithm by comparing the actual values with the values predicted by the model. It helps identify the types of errors made by the model and can be used to fine-tune the model's performance. The confusion matrix is organized into a grid with the actual classes on one axis (usually the y-axis) and the predicted classes on the other axis (usually the x-axis). Each cell in the matrix represents a combination of predicted and actual classes. This matrix provides the value for true positive, false positive, true negative, and false negative values which will be used in term to measeure the models accuracy, precision, recall and f1 scores.

ii. Accuracy

Accuracy quantifies the model's correctness in predicting true labels by comparing the total number of correctly classified instances of rice disease with the overall dataset size. It symbolizes the proportion of accurate predictions among all instances. A higher accuracy signifies fewer misclassifications, indicating the model's effectiveness in discerning between different classes of rice diseases. The accuracy equation is expressed as follows:

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative}} \quad \dots \dots \dots (1)$$

The accuracy metrics provides insights into the models' overall effectiveness in correctly classifying rice diseases. Furthermore, validation accuracy and training accuracy shed light on how well the models generalized to unseen data during the validation and training phases, respectively. Additionally, the loss metrics quantified the magnitude of errors incurred during the training process.

iii. Precision

Precision is a statistical measure that quantifies the accuracy of positive predictions made by a classification model. It calculates the proportion of true positive predictions among all instances predicted as positive by the model. Higher precision scores indicate greater reliability and accuracy in disease detection, while lower scores may suggest a higher rate of false positive predictions. Mathematically,

$$\text{Precision} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{False Positive}} \quad \dots \dots \dots (2)$$

iv. **Recall**

Recall is a statistical measure that quantifies the ability of a classification model to correctly identify all relevant instances of a particular class within a dataset. Higher recall scores indicate greater sensitivity and effectiveness in capturing true positive cases, while lower scores suggest potential shortcomings in detecting all instances of the disease. Mathematically,

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \dots\dots\dots (3)$$

v. **F1-Score**

F1 score combines precision and recall into a single metric, providing a balanced assessment of a classification model's performance. It is calculated as the harmonic mean of precision and recall, and it ranges between 0 and 1. A higher F1-score indicates perfect precision and recall of the proposed model.

$$\text{F1-Score} = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \dots\dots\dots (4)$$

vi. **Loss Calculation**

The test error percentage provides an indication of a model's accuracy in predicting outcomes on unseen data. A lower test error percentage suggests that the model is more accurate in its predictions, as it makes fewer mistakes when classifying instances in the test dataset. Since TensorFlow's `tf.keras.losses.SparseCategoricalCrossentropy` with `from_logits=False` is used for the loss calculation equation, it is based on the standard cross-entropy loss function for multi-class classification problems. Here's how the loss is calculated:

$$\text{Loss} = \frac{1}{N} \sum_{i=1}^N \log(y) \dots\dots\dots (5)$$

where,

N is the number of examples in the batch.

y is the predicted probability of the true class for the i -th example in the batch.

3.9 Implementation Tools

The implementation tools used for this study include:

1. Google Colab: Google Colab is a cloud-based platform that provides free access to GPU resources and allows for collaborative coding in Python. It enables users to run Python code in a Jupyter Notebook environment hosted on Google's servers, making it ideal for machine learning and data analysis tasks.

2. T4 GPU: The T4 GPU is a high-performance graphics processing unit developed by NVIDIA. It offers significant computational power, making it well-suited for training deep learning models and accelerating various machine learning tasks.

3. Keras: Keras is a Python-based high-level neural networks API that can operate on TensorFlow, Microsoft Cognitive Toolkit (CNTK), or Theano. It simplifies the creation and training of neural networks by offering a user-friendly interface, making the development of deep learning models more accessible.

3. TensorFlow: TensorFlow, a machine learning framework developed by Google, is open-source and encompasses a diverse array of tools, libraries, and resources for constructing and deploying machine learning models, including deep neural networks. Renowned for its adaptability, scalability, and superior performance, TensorFlow stands as one of the most prevalent frameworks in the realm of machine learning.

4. Python: Python is a widely used programming language known for its simplicity and readability. It offers extensive libraries and frameworks for various tasks, including machine learning, data analysis, and scientific computing. Python's versatility and ease of use make it the preferred choice for many machine-learning projects.

6. Matplotlib: Matplotlib is a plotting library for Python that enables users to create a wide range of static, interactive, and animated visualizations. It provides a flexible and customizable interface for generating plots, charts, and graphs, making it essential for data visualization and analysis tasks.

7. NumPy: NumPy is a fundamental package for scientific computing in Python. It provides support for multi-dimensional arrays, matrices, and mathematical functions, making it indispensable for numerical computations and data manipulation tasks in machine learning and data science projects.

These tools collectively offer a robust and efficient environment for developing, training, and evaluating machine learning models, facilitating the implementation of sophisticated algorithms and techniques in real-world applications.

3.10 Summary

The methodology section offers a thorough overview of the study's process, including model selection, transfer learning approach, dataset preparation, training, and testing phases. It justifies the rationale behind model selection, details the transfer learning framework employed, and outlines the workflow for dataset acquisition, preprocessing, and partitioning. The training and testing procedures are described, emphasizing key aspects such as optimizer selection, loss function definition, and prediction function.

CHAPTER 4

RESULT ANALYSIS & DISCUSSION

4.1 Introduction

The Results Analysis chapter serves as a comprehensive examination of the experimental outcomes obtained from the application of CNN, VGGNet16, ResNet50, and InceptionResNetV2 models in the context of rice disease detection. This chapter delves into an in-depth analysis of various performance metrics, including accuracy, precision, recall, and F1-Score, to assess the efficacy of each model in classifying rice diseases accurately. Furthermore, the chapter extends its evaluation beyond rice disease detection by exploring the performance of these models on other crops. Additionally, the chapter presents an evaluation of test error results, providing insights into the robustness and generalization capability of the models. By examining test error rates, the chapter offers valuable insights into the models' performance on unseen data, further enhancing the understanding of their reliability and effectiveness.

4.2 Performance of the Neural Network Models

In evaluating the performance of the neural networks - CNN, VGGNet16, ResNet50, and InceptionResNetV2—a comprehensive assessment is conducted. By thoroughly analyzing the confusion matrix and the derived metrics, it is possible to gain deep insights into the model's

performance and make informed decisions on potential improvements. Several important metrics such as accuracy, precision, recall, and f1 score can be derived from the confusion matrix to evaluate the performance of the classification model. Additionally, a comparative analysis of the training settings, including batch size, and optimizer used, sheds light on the efficiency and effectiveness of each model architecture.

4.2.1 VGGNet16

VGGNet16 achieved an accuracy of 98.44%, employing a batch size of 64 using the ‘rmsprop’ optimizer. We can observe that from the generated confusion matrix in Figure 4.1. Key matrices such as accuracy, precision, recall, and F1 Score can be derived from this matrix.

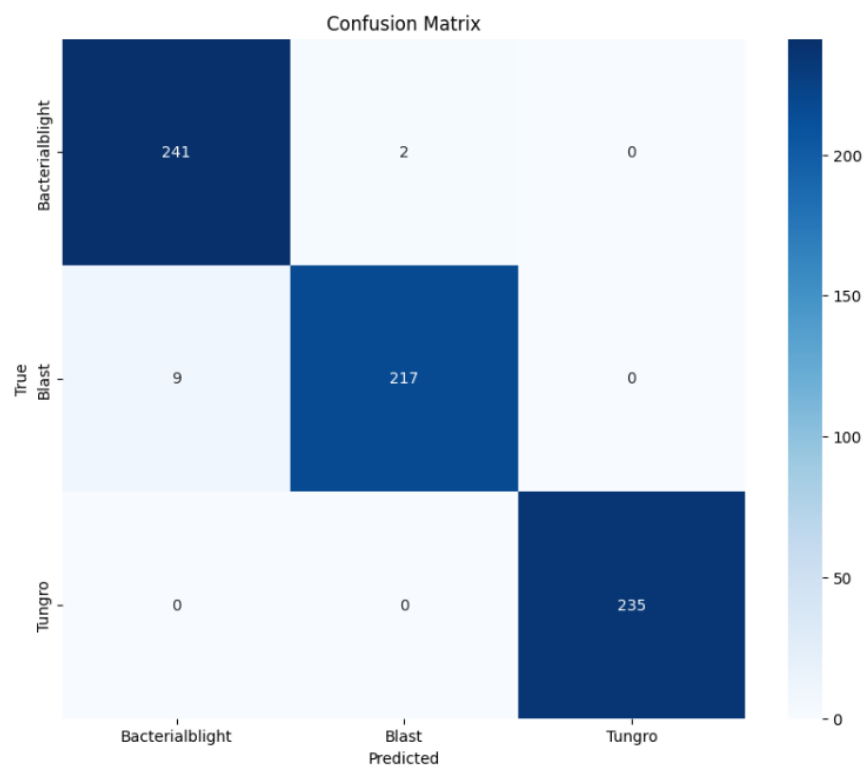


Figure 4.1 Confusion matrix for vggnet16

a) Accuracy

Total instances = $241 + 2 + 0 + 9 + 217 + 0 + 0 + 0 + 235 = 704$

Correct prediction = $241 + 217 + 235 = 693$

Accuracy = $\frac{693}{704} \approx 0.9844$ or 98.44%

b) Precision

$$\text{Precision}_{bb} = \frac{241}{241+9} \approx 0.964$$

$$\text{Precision}_{bl} = \frac{217}{217+2} \approx 0.991$$

$$\text{Precision}_{tg} = \frac{235}{235+0} \approx 1.00$$

Precision is also very high for all classes, with Tungro achieving a perfect precision score of 1.000, indicating no false positives for this class.

c) Recall

$$\text{Recall}_{bb} = \frac{241}{241+2} \approx 0.992$$

$$\text{Recall}_{bl} = \frac{217}{217+9} \approx 0.960$$

$$\text{Recall}_{tg} = \frac{235}{235+0} \approx 1.00$$

Recall is equally high, with Tungro again achieving a perfect recall score of 1.000, indicating no false negatives for this class. Bacterial Blight and Blast have slightly lower recall scores but are still very high.

d) F1 Score

$$\text{F1 Score}_{bb} = 2 \times \frac{0.964 \times 0.992}{0.964 + 0.992} \approx 0.978$$

$$\text{F1 Score}_{bl} = 2 \times \frac{0.991 \times 0.960}{0.991 + 0.960} \approx 0.975$$

$$\text{F1 Score}_{tg} = 2 \times \frac{1.00 \times 1.00}{1.00 + 1.00} \approx 1.00$$

The F1 scores for all classes are excellent, with Tungro having a perfect F1 score of 1.000, indicating the model's exceptional performance in predicting this class. Bacterial Blight and Blast also have high F1 scores, indicating good balance between precision and recall.

The model performs exceptionally well, especially for detecting Tungro. However, there's a slight room for improvement in detecting Blast, where the recall is the lowest (though still high). Most misclassifications occur between Bacterial Blight and Blast. This could be due to similar visual features between these two diseases.

4.2.2 ResNet50

ResNet50 achieved an accuracy of 50.99% over the course of 10 and 30 epochs, employing a batch size of 64 in all the optimizers. From the generated confusion matrix in Figure 4.2, we can derive key metrics such as accuracy, precision, recall, and F1 score.

a) Accuracy

$$\text{Total instances} = 138 + 73 + 54 + 19 + 180 + 37 + 6 + 156 + 41 = 704$$

$$\text{Correct prediction} = 138 + 180 + 41 = 359$$

$$\text{Accuracy} = \frac{359}{704} \approx 0.510 \text{ or } 51.0\%$$

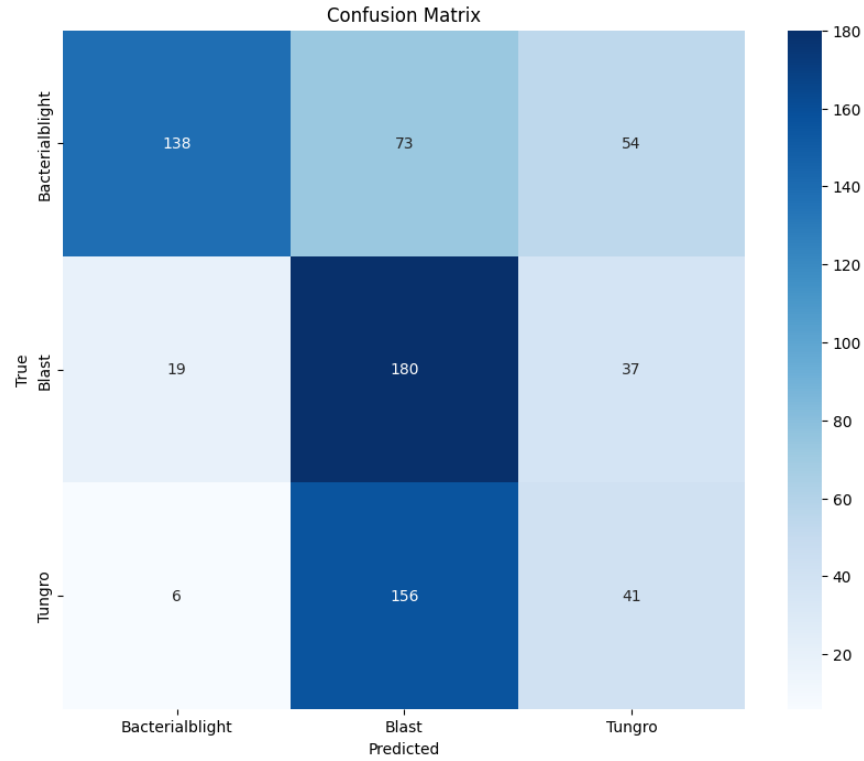


Figure 4.2 Confusion matrix for resnet50

b) Precision

$$\text{Precision}_{bb} = \frac{138}{138 + 19 + 6} \approx 0.847$$

$$\text{Precision}_{bl} = \frac{180}{73 + 180 + 156} \approx 0.440$$

$$\text{Precision}_{tg} = \frac{41}{54 + 37 + 41} \approx 0.311$$

Precision varies significantly across the classes. Bacterial Blight has a high precision (0.847), indicating that when the model predicts Bacterial Blight, it is usually correct. Blast provides a moderate precision (0.440), suggesting a relatively high number of false positives. Tungro has the lowest precision (0.311), indicating a high number of false positives.

c) Recall

$$\text{Recall}_{bb} = \frac{138}{138 + 73 + 54} \approx 0.521$$

$$\text{Recall}_{bl} = \frac{180}{180 + 19 + 37} \approx 0.763$$

$$\text{Recall}_{tg} = \frac{41}{41 + 6 + 156} \approx 0.202$$

Recall varies significantly across the classes. Bacterial Blight has a moderate recall of 0.521, indicating that the model misses nearly half of the actual Bacterial Blight cases. Blast has a high recall of 0.763, suggesting the model successfully identifies most Blast cases. Tungro, however, has a low recall of 0.202, indicating the model misses most of the Tungro cases.

d) F1 Score

$$\text{F1 Score}_{bb} = 2 \times \frac{0.847 \times 0.521}{0.847 + 0.521} \approx 0.644$$

$$\text{F1 Score}_{bl} = 2 \times \frac{0.440 \times 0.763}{0.440 + 0.763} \approx 0.558$$

$$\text{F1 Score}_{tg} = 2 \times \frac{0.311 \times 0.202}{0.311 + 0.202} \approx 0.247$$

The F1 scores for all classes are excellent, with Tungro having a perfect F1 score of 1.000, indicating the model's exceptional performance in predicting this class. Bacterial Blight and Blast also have high F1 scores, indicating good balance between precision and recall.

The overall accuracy of the model is 51.0%, which indicates that the model correctly classifies about half of the instances. This is relatively low, suggesting the model has room for improvement. The model's performance is quite low for Tungro and moderate for Blast.

4.2.3 InceptionResNetV2

InceptionResNetV2 obtained an accuracy of 98.21% throughout both 10 and 30 epochs, employing a batch size of 64. From the generated confusion matrix in Figure 4.3, we can derive key metrics such as accuracy, precision, recall, and F1 score.

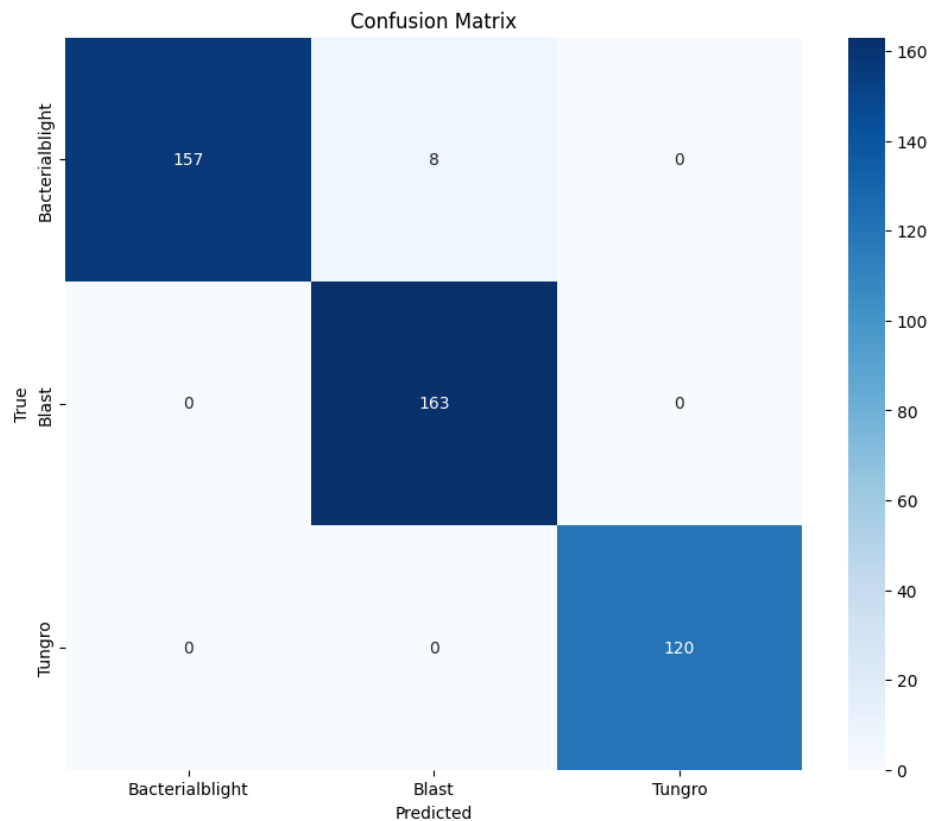


Figure 4.3 Confusion matrix for inceptionresnetv2

a) Accuracy

Total instances = $157 + 8 + 0 + 0 + 163 + 0 + 0 + 0 + 120 = 448$

Correct prediction = $157 + 163 + 120 = 440$

$$\text{Accuracy} = \frac{448}{440} \approx 0.982 \text{ or } 98.2\%$$

b) Precision

$$\text{Precision}_{bb} = \frac{157}{157+0} \approx 1.00$$

$$\text{Precision}_{bl} = \frac{163}{8+163} \approx 0.953$$

$$\text{Precision}_{tg} = \frac{120}{0+120} \approx 1.00$$

Precision is also very high for all classes, with Tungro and Blast achieving a perfect precision score of 1.000, indicating no false positives for these classes.

c) Recall

$$\text{Recall}_{bb} = \frac{157}{157+8} \approx 0.9521$$

$$\text{Recall}_{bl} = \frac{163}{163+0} \approx 1.00$$

$$\text{Recall}_{tg} = \frac{120}{120+0} \approx 1.00$$

Recall is equally high, with Tungro and Blast again achieving a perfect recall score of 1.000, indicating no false negatives for these classes. Bacterial Blight has a slightly lower recall scores but is still very high.

d) F1 Score

$$\text{F1 Score}_{bb} = 2 \times \frac{1.00 \times 0.951}{1.00 + 0.951} \approx 0.975$$

$$\text{F1 Score}_{\text{bl}} = 2 \times \frac{0.953 \times 1.00}{0.953 + 1.00} \approx 0.976$$

$$\text{F1 Score}_{\text{tg}} = 2 \times \frac{1.00 \times 1.00}{1.00 + 1.00} \approx 1.00$$

The F1 scores are very high across all classes, with Bacterial Blight at 0.975, Blast at 0.976, and Tungro at 1.000. This demonstrates that the model has a good balance between precision and recall for all classes, particularly excelling in predicting Tungro and Blast.

The model performs exceptionally well across all classes, with near-perfect precision, recall, and F1 scores, and a high overall accuracy. The only slight area for improvement is the recall for Bacterial Blight, which, while still very high, could be optimized further.

4.3 Optimizer

Three optimizers namely ‘adam’, ‘sgd’, and ‘rmsprop’ were utilized in the ImageNet pre-trained models. Among them ‘rmsprop’ provided the highest accuracy.

4.3.1 Accuracy Calculation

The analysis of three models (VGGNet16, ResNet50, and InceptionResNetV2) with three different optimizers (adam, sgd, and rmsprop) for rice disease detection is shown in Figure 4.4. VGGNet16 achieves the highest accuracy with rmsprop at 98.44%, suggesting this optimizer significantly boosts its performance. InceptionResNetV2 also performs exceptionally well with rmsprop, achieving 98.21%, and maintains high accuracy across all optimizers, indicating its robustness and effectiveness for the task. In contrast, ResNet50 shows relatively low performance across all optimizers, with the highest accuracy of 50.99% using rmsprop, suggesting it may require further tuning or may not be the best model choice. For optimal results, InceptionResNetV2 with rmsprop is recommended, while VGGNet16 with rmsprop is also a strong alternative.

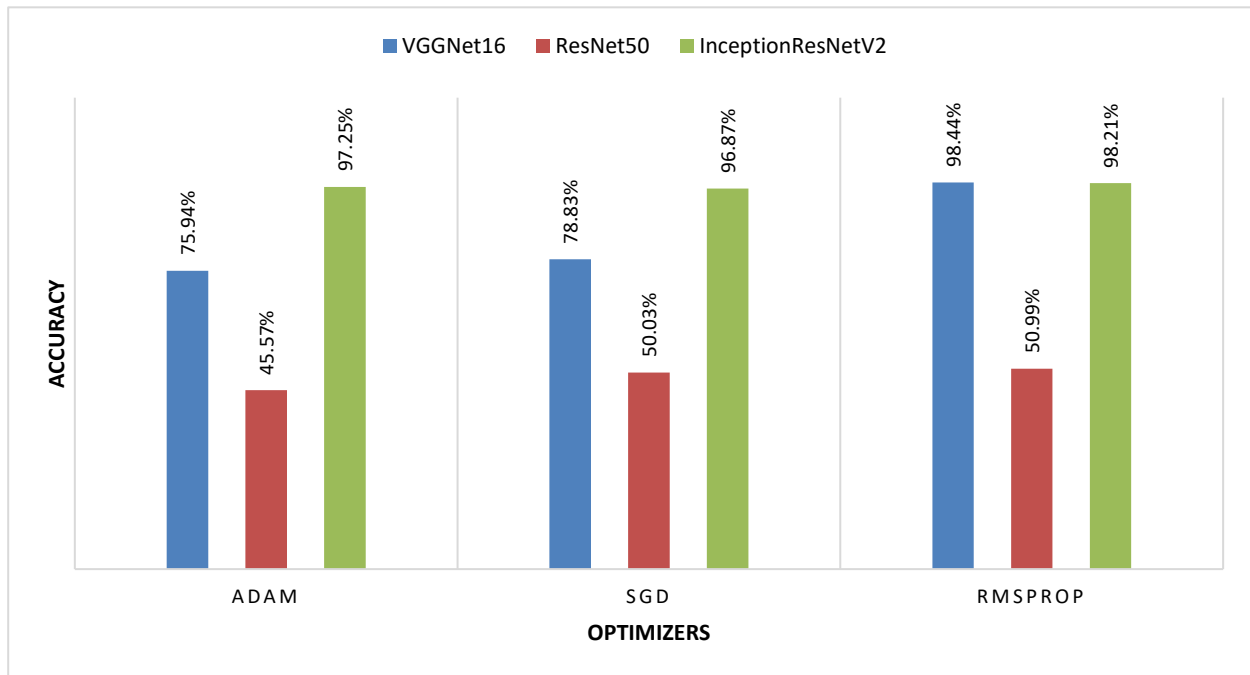


Figure 4.4 Performance analysis for different optimizers

For a comparative analysis to show the need for using optimizer, this study also tests the models without implementing the optimizers which is demonstrated in Figure 4.5

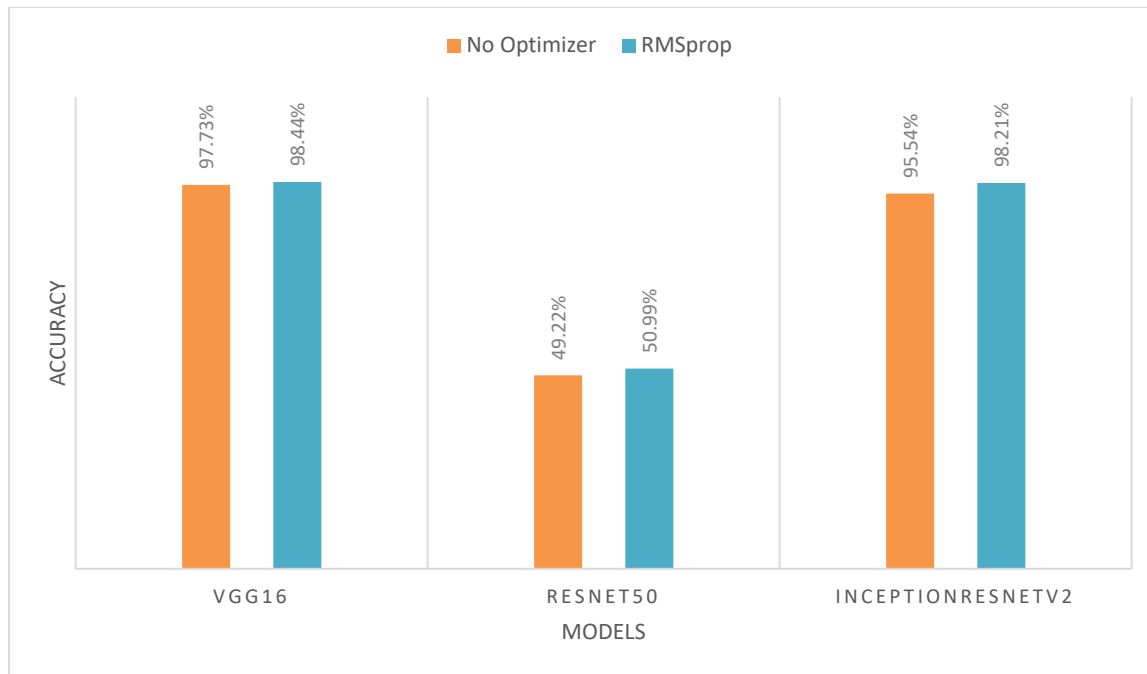


Figure 4.5 Impact of Optimizer Usage on Model Performance

By comparing these results, it is evident that it's essential to use an optimizer during the training of neural networks to optimize the model's weights and biases and minimize the loss function.

4.3.2 Error Analysis

The summary table for test error scores is shown in Table 4.1. This error rate is based on using the rmsprop optimizer.

Table 4.1 Error Summary

	CNN	VGGNet16	ResNet50	InceptionResNetV2
Error	11.91%	0.12%	46.67%	0.61%

VGG16 emerges as the most reliable model for rice disease detection, given its significantly lower test error rate compared to CNN, ResNet50, and InceptionResNetV2. It demonstrates superior performance in accurately classifying rice disease instances, making it a promising choice for practical applications in agricultural settings. ResNet50 presented a comparatively higher test error rate of 46.67%. Challenges or limitations in the model's architecture or training process may lead to a higher test error rate.

4.4 Comparative Analysis

In this section, our primary aim is to conduct a thorough comparative analysis of the models employed for rice disease detection. Our goal is to determine the model that best suits our specific problem. To accomplish this objective, we will assess each model using a range of classification metrics. Through this comparative evaluation, we aim to discern the most dependable model for accurately detecting rice diseases. Table 4.6 gives an overview of the performance metrics

Table 4.2 Models Comparison Parameters

Models	Accuracy	Precision	Recall	F1-Score
CNN	97.66%	94.10%	93.86%	93.91%
VGGNet16	98.44%	98.46%	98.43%	98.44%
ResNet50	50.99%	55.57%	50.97%	50.04%
InceptionResNetV2	98.21%	98.29%	98.21%	98.21%

VGGNet16 and InceptionResNetV2 outperform CNN and ResNet50 in terms of accuracy, precision, recall, and F1 score. VGGNet16 stands out with the highest accuracy and precision, while InceptionResNetV2 demonstrates consistently high performance across all metrics.

ResNet50 lags with comparatively lower accuracy and F1-score, indicating less reliable performance in rice disease detection.

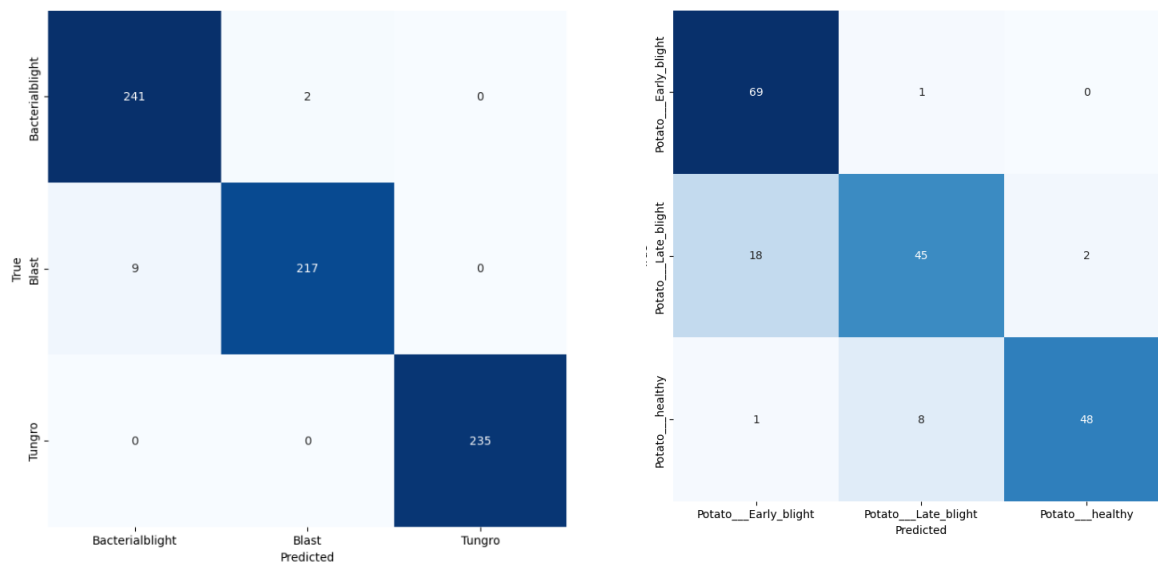
4.5 Testing for different Crop

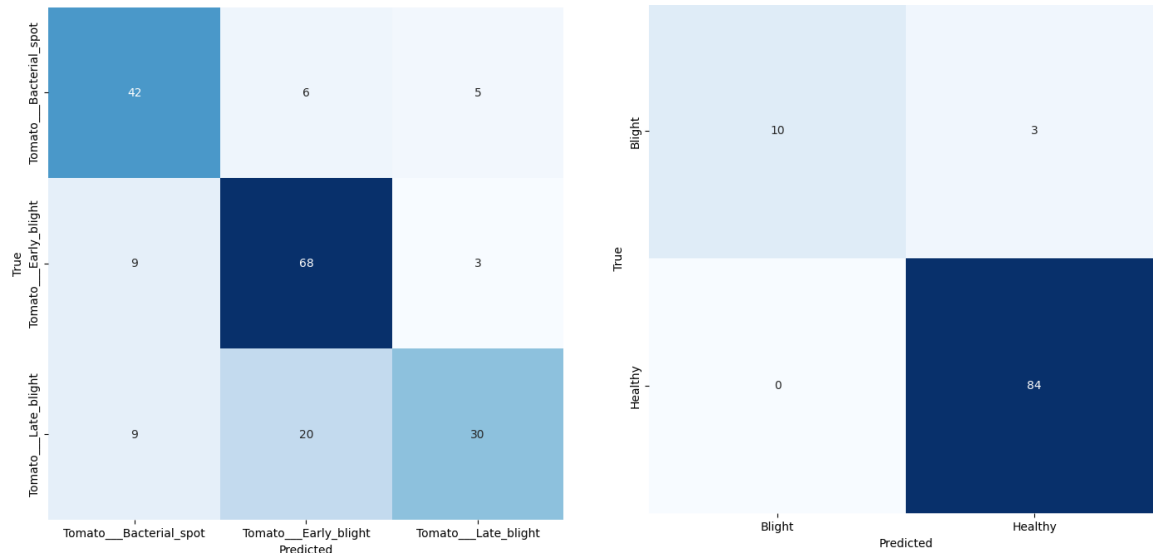
These models were also used for three different crops: potato, tomato, and corn. There were three diseases for potato, three disease classes for tomato, one disease class for corn, and one healthy corn leaf class. They were all trained using the ‘rmsprop’ optimizer since it demonstrated the best results for the main crop of this study.

a) VGGNet16

The VGGNet16 model performs best on Rice and Corn datasets with high True Positive values. Performance is relatively lower for Potato and Tomato, particularly for the Late Blight class in Tomato. A comparison on the confusion matrices for these crops is provided in Table 4.3

Table 4.3 Confusion matrix for different crops using vggnet16

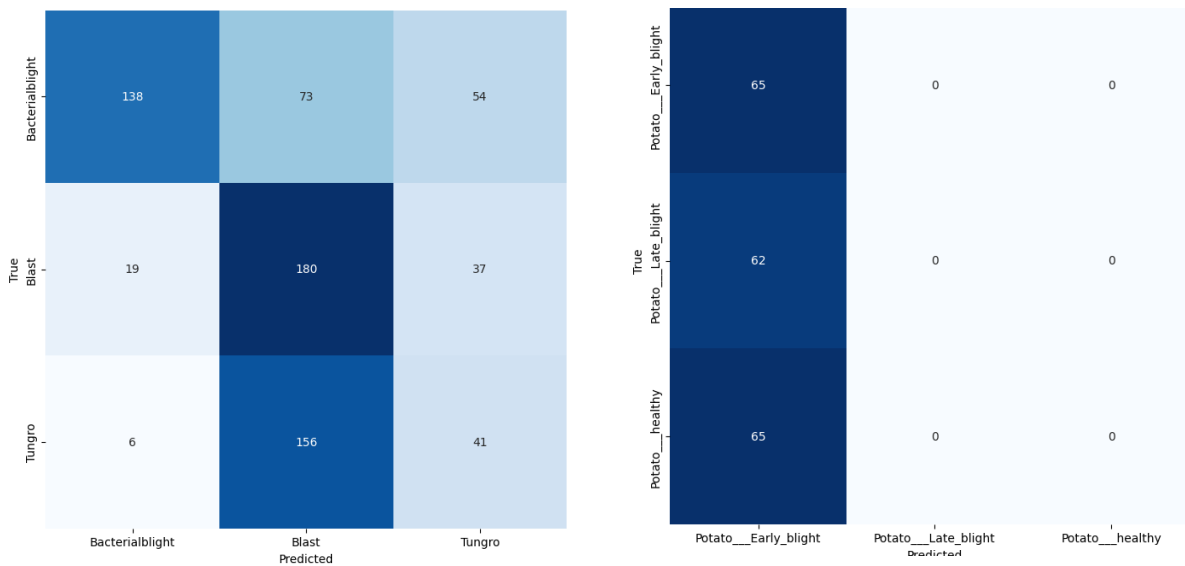


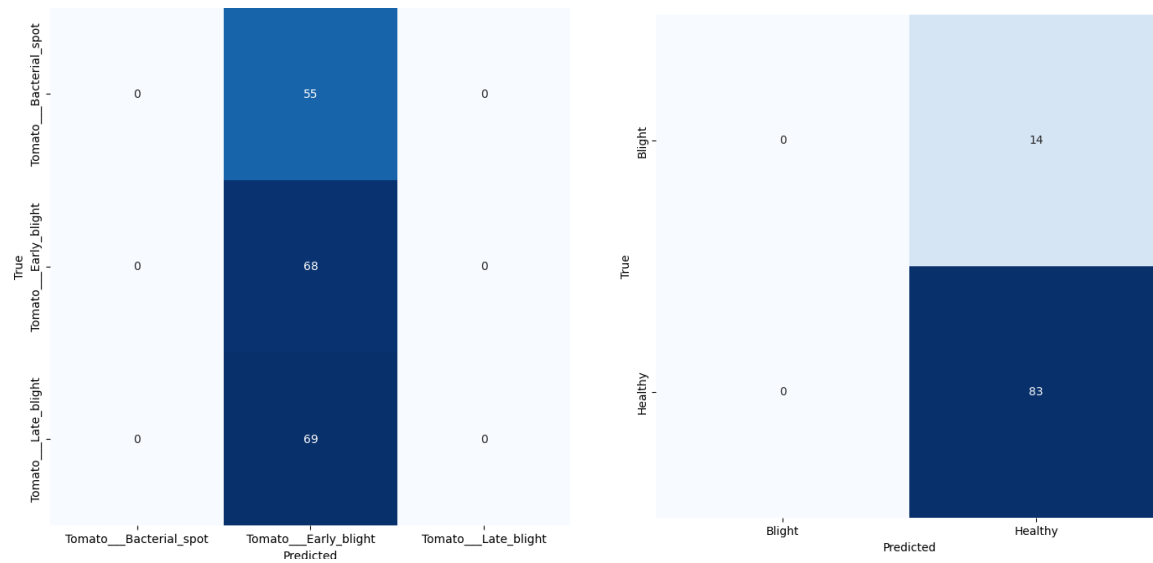


b) ResNet50

The provided confusion matrices in Table 4.4 for rice, potato, tomato, and corn represent the performance of a ResNet50 model in classifying diseases specific to each plant.

Table 4.4 Confusion matrix for different crops using resnet50



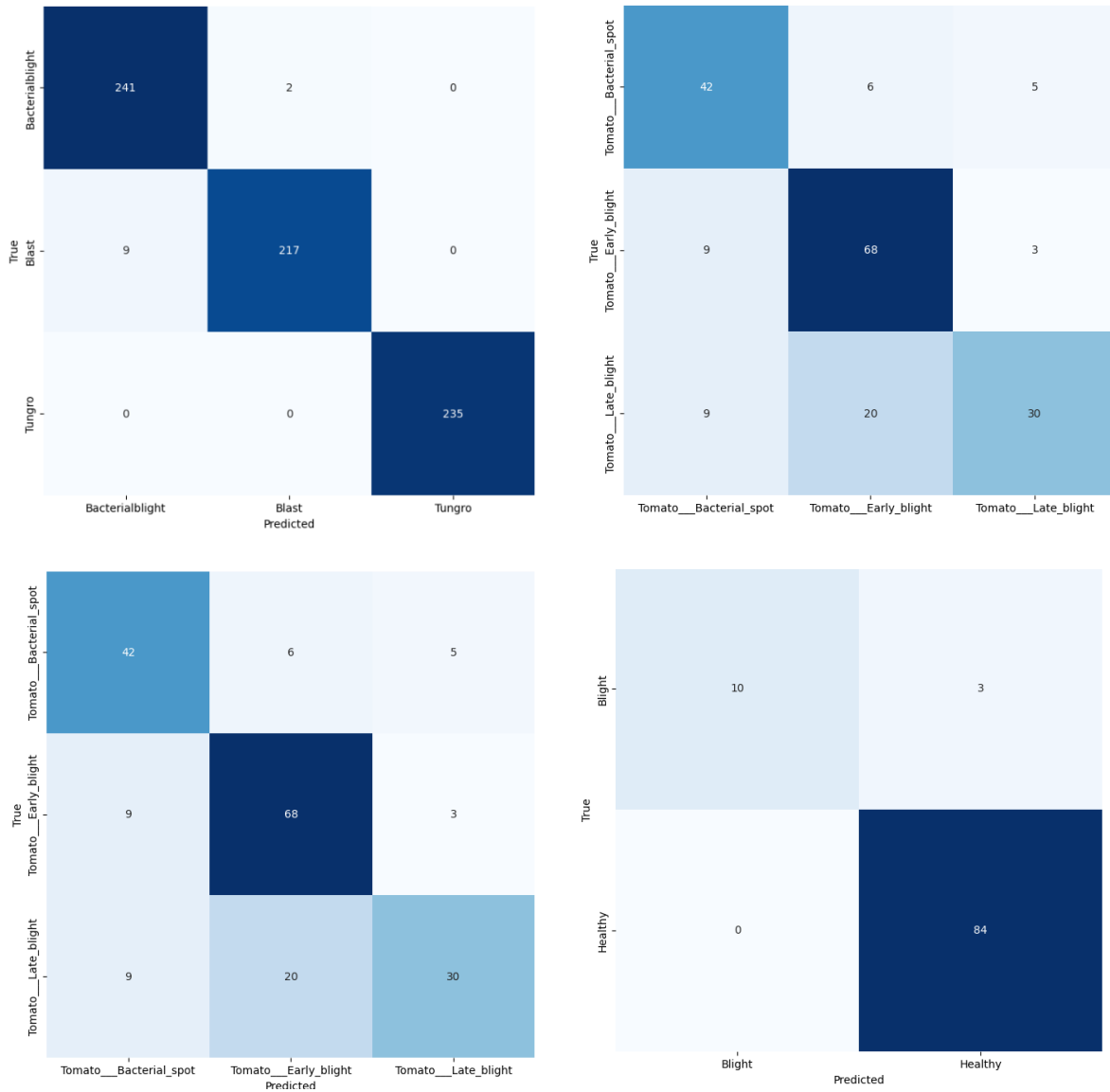


The model shows the highest accuracy in detecting Healthy Corn and Early Blight and Late Blight in Tomatoes. The confusion matrices indicate severe class imbalance or dataset issues, particularly for Potato and Corn, where certain classes are never predicted correctly.

c) InceptionResNetV2

The InceptionResNetv2 model shows outstanding performance in classifying diseases in Rice, Healthy potatoes, Bacterial Spot in Tomatoes, and Corn. The model demonstrates a significant improvement over the previous ResNet50 model, particularly in achieving higher accuracy rates across most categories.

Table 4.5 Confusion matrix for different crops using inceptionresnetv2



The accuracy summary of vggnet16, resnet50, and inceptionetv2 on these crops are shown in Table 4.6

Table 4.6 Accuracy summary for different crops

Models	Rice	Potato	Tomato	Corn
--------	------	--------	--------	------

VGGNet16	98.86%	84.38%	72.92%	96.91%
ResNet50	50.99%	33.85%	35.42%	85.57%
InceptionResNetV2	98.21%	90.62%	78.12%	96.88%

The results obtained after training the models originally designed for rice disease detection on datasets for other crops imply their transferability and generalization capability across different agricultural contexts. Despite being trained on rice disease data, these models demonstrate high accuracy in detecting diseases in other crops, suggesting that they have learned meaningful and robust features applicable to various plant species. However, fine-tuning or retraining the models using domain-specific data for each crop could further enhance their performance and adaptability. Overall, these findings highlight the potential of these models as valuable diagnostic tools for farmers, contributing to improved crop health and yield across diverse agricultural settings.

4.6 Comparison with Existing Works

Table 4.7 compares the accuracy of previous proposed work on different plant disease detection methods to the results of this study.

Table 4.7 Comparison with existing works

Authors	Year	Crop	Methods	Accuracy
Ghoshal et al.[17]	2020	Rice	VGG	92.46%
Wei-Jian et al. [27]	2020	Different crops	MDFC-ResNet	84.22%
Coulibaly et al. [34]	2019	Millet	VGG16	95%%

Ferentinos et al. [18]	2018	25 crops	VGG	99.53%
Xie et al. [26]	2020	Grape	InceptionResNet	81.1%
Kathiresan et al.[22]	2021	3 different crops	RiceDenseNet	98.78%
Sethi et al. [23]	2020	Rice	ResNet50	98.38%
Sharma et al. [24]	2020	Rice	VGG-16	91%
Proposed Model	2024	Rice	VGG16	98.44%

4.7 Summary

The Results Analysis chapter serves as a comprehensive examination of the experimental findings, offering a detailed exploration of the performance and applicability of the CNN, VGGNet16, ResNet50, and InceptionResNetV2 models in rice disease detection and beyond.

CHAPTER 5

CONCLUSION

5.1 Conclusion

In conclusion, the evaluation of CNN, VGGNet16, ResNet50, and InceptionResNetV2 models for rice disease detection revealed significant insights into their performance and efficacy. VGGNet16 and InceptionResNetV2 emerged as the top-performing models, exhibiting high accuracy of 98.44% and 98.21% respectively. Their robust performance underscores their potential as reliable tools for timely and accurate disease detection in rice crops. Conversely, ResNet50 demonstrated comparatively lower performance metrics achieving an accuracy of 50.21%, indicating limitations in effectively capturing and classifying disease features within the dataset.

The choice of optimizer, particularly RMSprop, significantly influenced the performance of VGGNet16, highlighting the importance of optimization strategies in model training. Additionally, the transferability of these models to other crops underscores their versatility and potential impact in diverse agricultural contexts. However, further fine-tuning and retraining may be necessary to maximize their performance for specific crop types.

5.2 Future Scope

The future scope of this study encompasses several promising avenues for further research and application, while also acknowledging the limitations encountered during the research process:

- 1. Enhanced Model Architectures:** Exploring more advanced neural network architectures and techniques, such as attention mechanisms or graph convolutional networks and advanced data augmentation techniques, such as generative adversarial networks (GANs) or self-supervised learning, can further enhance model performance and alleviate data scarcity challenges.
- 2. Fine-Tuning and Transfer Learning:** Further investigations into fine-tuning pre-trained models on domain-specific datasets and exploring transfer learning techniques tailored to different crops and diseases could optimize model performance for specific agricultural contexts.
- 3. Real-Time Deployment:** Developing real-time deployment strategies for trained models, such as integrating them into smartphone applications or IoT devices, can enable farmers to quickly and accurately diagnose rice diseases in the field, facilitating timely interventions.
- 4. Interdisciplinary Collaboration:** Collaborating with agronomists, plant pathologists, and agricultural experts can provide valuable domain knowledge and insights, fostering the development of more effective and practical disease detection solutions.

Summary

In conclusion, while this study provides valuable insights into the application of deep learning models for rice disease detection, several limitations should be acknowledged. The size and diversity of the dataset, the need for comprehensive evaluation metrics, and the challenges of real-world deployment are critical areas that require further research and development. By addressing these limitations and exploring the outlined future directions, the field of rice disease detection can continue to advance, ultimately contributing to improved agricultural productivity and food security.

REFERENCES

- [1] Al-Mamun M. A. 2021. Growth and Trend Analysis of Area, Production and Yield of Rice: A Scenario of Rice Security In Bangladesh. PLOS ONE. Vol. 16.
- [2] Alatawi, A. A., Alomani, S. M. , Alhawiti, N. I., and Ayaz. M. 2022. Plant Disease Detection using AI based VGG-16 Model. International Journal of Advanced Computer Science and Applications. Vol. 13.
- [3] Jiang, F., Lu, Y., Chen, Y., Cai, D., and Li, G. 2020. Image recognition of four rice leaf diseases based on deep learning and support vector machine. Comput Electron Agric. Vol. 179.
- [4] Applalanaidu, M. V., and Kumaravelan, G. 2021. A review of machine learning approaches in plant leaf disease detection and classification. Institute of Electrical and Electronics Engineers Inc.
- [5] Shrivastava, V. K., Pradhan, M. K., Minz, S., and Thakur, M. P. 2019. Rice plant disease classification using transfer learning of deep convolution neural network. International Society for Photogrammetry and Remote Sensing.
- [6] Leonard, A. and Gianessi, P. 2014. Importance of Pesticides for Growing Rice in South and South East.
- [7] Liu, L. M., Zhao, Y., Zhang, Y. L., Wang, L., Hou, Y. X., and Huang, S. W. First Report of Leaf Spot Disease on Rice Caused by *Epicoccum sorghinum* in China.
- [8] Mathulapransan, S., Lanthong, K., and Patarapuwadol, S. 2020. *Rice Diseases Recognition Using Effective Deep Learning Models*.
- [9] Dastogeer, K. M. G. Study of Rice Diseases
- [10] Akter, T., Parvin, M. T., Mila, F. A., and Nahar, A. 2019. Factors determining the profitability of rice farming in Bangladesh. *Journal of the Bangladesh Agricultural University*. Vol. 17.
- [11] Khatun, M., Nessa, B., Salam, M., and Kabir, M. 2021. Strategy for Rice Disease Management in Bangladesh. *Bangladesh Rice Journal*. Vol. 25.

- [12] Parveen, S., Ali, M., and Ali, M. 2019. Screening Rice Germplasm against Sheath Blight Disease of Rice and its Integrated Management in Bangladesh.. *Bangladesh Rice Journal*. Vol. 22
- [13] Rashid, M.M., Mian, M., Hossain, M., Islam, A., Nihad, S.A.I., Khan, M.A.I., Ansari, TH., Latif, M.A.2021. Factors Affecting Rice Tungro Disease and Its Management in Cumilla Region.
- [14] Kaggle: Your Home for Data Science.
- [15] Szegedy C. 2014. Going Deeper with Convolutions.
- [16] DenseNet Explained. Papers With Code.
- [17] Ghosal, S., and Sarkar, K. 2020. *Rice Leaf Diseases Classification Using CNN With Transfer Learning*. Institute of Electrical and Electronics Engineers.
- [18] Ferentinos, K. P. 2018. Deep learning models for plant disease detection and diagnosis. *Comput Electron Agric*. Vol. 145
- [19] Islam, M. A., Rahman, N., Shamsoddin, M., Hasan, S., Hossain, S. and Khatun, T. 2021. An Automated Convolutional Neural Network Based Approach for Paddy Leaf Disease Detection. *International Journal of Advanced Computer Science and Applications*. Vol. 1.
- [20] Chen, J., Chen, J., Zhang, D., Sun, Y., and Nanekaran, Y.A. 2020. Using deep transfer learning for image-based plant disease identification. *Comput Electron Agric*. Vol. 173
- [21] Chen, J., Zhang, D., Nanekaran, Y. A., and Li, D. 2020. Detection of rice plant diseases based on deep transfer learning. *J Sci Food Agric*. Vol. 100
- [22] Kathiresan, G., Anirudh, M., Nagharjun, M., and Karthik, R. 2021. Disease detection in rice leaves using transfer learning techniques. *Journal of Physics: Conference Series*.
- [23] Sethy, P. K., Barpanda, N. K., Rath, A. K., and Behera, S. K. 2020. Deep feature based rice leaf disease identification using support vector machine. *Comput Electron Agric*, Vol. 175.
- [24] Sharma, R., Das, S., Gourisaria, M. K., Rautaray, S. S., and Pandey, M. 2020. A Model for Prediction of Paddy Crop Disease Using CNN. *Advances in Intelligent Systems and Computing*. Springer.
- [25] Singh, V., and Misra, A. K. 2017. Detection of plant leaf diseases using image segmentation and soft computing techniques. *Information Processing in Agriculture*, Vol. 4.

- [26] Xie, X., Ma, Y., Liu, B., He, J., Li, S., and Wang, H. 2020. A Deep-Learning-Based Real-Time Detector for Grape Leaf Diseases Using Improved Convolutional Neural Networks. *Front Plant Sci.* Vol. 11.
- [27] Hu, W. J., Fan, J., Du, X., Li, B. S., Xiong, N., and Bekkering, E. 2020. MDfC-ResNet: An Agricultural IoT System to Accurately Recognize Crop Diseases. *IEEE.* Vol. 8.
- [28] Temniranrat, P., Kiratiratanapruk, K., Kitvimonrat, A., Sinthupinyo, W., and Patarapuwadol, S. 2021. A system for automatic rice disease detection from rice paddy images serviced via a Chatbot. *Comput Electron Agric.* Vol. 185.
- [29] Brahim M. 2018. Deep Learning for Plant Diseases: Detection and Saliency Map Visualisation Machine learning approaches to rank news feed updates on social media View project Geospatial Decision Making approaches View project Deep Learning For Plant Diseases: Detection and Saliency map Visualization.
- [30] Brahim, M., Boukhalfa, K., and Moussaoui, A. 2017. Deep Learning for Tomato Diseases: Classification and Symptoms Visualization. *Applied Artificial Intelligence.* Vol. 31.
- [31] Peng, J., Wang, Y., Jiang, P., Zhang, R., and Chen, H. 2023. RiceDRA-Net: Precise Identification of Rice Leaf Diseases with Complex Backgrounds Using a Res-Attention Mechanism. *Applied Sciences (Switzerland).* Vol. 13.
- [32] Bashir, K., Rehman, M., and Bari, M. 2019. Detection and Classification of Rice Diseases: An Automated Approach Using Textural Features. *Mehran University Research Journal of Engineering and Technology.* Vol. 38.
- [33] Optimizers. <https://keras.io/api/optimizers/>.
- [34] Coulibaly, S., Kamsu-Foguem, B., Kamissoko, D., and Traore, D. 2019. Deep neural networks with transfer learning in millet crop images. *Comput Ind.* Vol. 108.