Table of Contents

**Topic**   **Page No**

Abstract………………………………………………...………………………………………… I

Table of Content…………………………………………...……………………………………. II

List of Figures………………………………………...………………………………………... III

List of Tables……………………………………………...…………………………………….IV

List of Equations……………………………………………………………….…………….….V

[**CHAPTER 1:INTRODUCTION** 1](#_Toc186375349)

1.1 Rice in Agriculture………………………………………………………………………….1

[1.2 Challenges Of Rice Leaf Diseases 2](#_Toc186375352)

1.3 Limitation of Traditional Disease Detection Methods………………………...……………2

[1.4 Role Of Artificial Intelligence In Disease Identification 4](#_Toc186375354)

[1.5](#_Toc186375355) Integrating Ai And Dmd For Improved Disease Detection………………………………....3

[1.6 Motivation 5](#_Toc186375356)

[**CHAPTER 2: LITERATURE SURVEY** 6](#_Toc186375357)

[2.1 Overview 6](#_Toc186375359)

[2.2 Literature Review 6](#_Toc186375360)

[2.3 Research Gap 9](#_Toc186375361)

[2.4 Problem Statement 9](#_Toc186375362)

[2.5 Objectives 10](#_Toc186375363)

[**CHAPTER 3: REQUIREMENT ANALYSIS** 11](#_Toc186375364)

[3.1 Functional Requirements 11](#_Toc186375366)

[3.2 Non-Functional Requirements 11](#_Toc186375367)

[3.3 Hardware Components 12](#_Toc186375368)

[3.4 Software Components 12](#_Toc186375369)

[**CHAPTER 4: DATASET DESCRIPTION** 15](#_Toc186375372)

[4.1 Data Overview: 15](#_Toc186375374)

[4.2 Dataset Split: 17](#_Toc186375375)

[**CHAPTER 5: METHODOLOGY** 19](#_Toc186375376)

[5.1 Methodology Overview 19](#_Toc186375378)

[5.2 Data Collection And Dataset Preparation 20](#_Toc186375379)

[5.3 Dynamic Mode Decomposition (DMD) For Preprocessing 22](#_Toc186375380)

[5.4. Densenet121 For Feature Extraction And Classification 27](#_Toc186375381)

[5.5 Advantages of DenseNet121 30](#_Toc186375382)

[**CHAPTER 6: IMPLEMENTATION** 31](#_Toc186375383)

[6.1 Data Collection And Preprocessing 31](#_Toc186375385)

[6.2 Dynamic Mode Decomposition (DMD) Preprocessing 31](#_Toc186375386)

[6.3 Densenet121 Model Implementation 32](#_Toc186375390)

[6.4framework And Libraries 33](#_Toc186375391)

[6.5 Training The Model 33](#_Toc186375392)

[6.6 Evaluation 35](#_Toc186375393)

[**CHAPTER 7: RESULTS AND ANALYSIS** 37](#_Toc186375394)

[7.1 Dynamic Mode Decomposition (DMD) Results 37](#_Toc186375396)

[7.2 Key Observations 37](#_Toc186375397)

[7.3 Quantitative Results 38](#_Toc186375398)

[7.4.Analysis Of The Results 40](#_Toc186375399)

[**CHAPTER 8: CONCLUSION AND FUTURE SCOPE** 42](#_Toc186375400)

[**REFERENCES** 43](#_Toc186375402)

**List of Figures**

Figure 1.1: A Healthy Rice Plant ..................................................................................................1

Figure 1.2: Bacterial Blight ..........................................................................................................2

Figure 1.3: Leaf Blast ...................................................................................................................2

Figure 1.4: Brown Spot ................................................................................................................3

Figure 1.5: Tungro ........................................................................................................................3

Figure 1.6: Hispa ..........................................................................................................................4

Figure 4.1: Sample Dataset Images ............................................................................................16

Figure 5.1: High-Level Design ..................................................................................................19

Figure 5.2: The DMD-Based ROI Preprocessing Architecture……………………………….23

Figure 5.3: Color channel representation of rice leaf images ………………………………… 26

Figure 5.4: DenseNet121 Architecture……………………………………………………….. 27

Figure 6.1:Sample Rice leaf images and the results from DMD based processing……………32

Figure 7.1: Results After Dynamic Mode Decomposition .............. …………………………. 38

Figure 7.2: Graph Representing Accuracy for All the Classes ..................................................39

Figure 7.3: Confusion Matrix for All the Diseases ...................................................................40

**LIST OF TABLES**

Table 1: Literature Survey ....................................................................................................…..6

Table 2: Class-Specific Metrics ..........................................................................................….39

**LIST OF EQUATIONS**

Equation 1: Snapshot Matrix Construction ..............................................................................24

Equation 2: Shifted Snapshot Matrices .............................. ………………………………….24

Equation 3: Koopman Operator Representation ......................................... …………………24

Equation 4: Computation of Koopman Operator ....................................... ………………....24

Equation 5: Computation of Koopman Operator using pseudo inverse ..... …………………24

Equation 6: Singular Value Decomposition (SVD) ................................... …………………25

Equation 7: Reduced Koopman Operator ............................. …………………………… …25

Equation 8: Dynamic Modes and Sparse Components ............................. ………………. …25

Equation 9: Accuracy ..............................................................................................................35

Equation 10: Precision .............................................. ……………………………………….36

Equation 11: Recall(Sensitivity) .............................................. …………………………….36

Equation 12: F1 Score.............................................. ……………………………………….36

# CHAPTER 1

# INTRODUCTION

This chapter provides a comprehensive overview of the research context and introduces the key elements of the project. It covers the significance of rice cultivation, the challenges of rice leaf diseases, and the limitations of traditional disease detection methods. It also highlights the role of Artificial Intelligence (AI) in improving disease detection and how Dynamic Mode Decomposition (DMD) is integrated to enhance disease identification accuracy and infected region segmentation. The project utilizes the DenseNet121 model for disease classification, combined with DMD to achieve better results.

**1.1 Rice in Agriculture**

Rice is one of the most important staple food crops, supporting over 3.5 billion people worldwide. It is particularly significant in Asia, where most global rice production occurs. In India, rice cultivation is a vital economic activity, especially in the southern states such as Andhra Pradesh, Tamil Nadu, Kerala, and Karnataka. Despite rice's critical role in food security, its productivity is threatened by various pests and diseases, which can lead to substantial yield losses if not identified and managed early. The rice plant is illustrated in Figure 1.1, providing a visual representation of a healthy rice plant.



Figure 1.1: A Healthy Rice Plant

## 

## 1.2 Challenges of Rice Leaf Diseases

Rice plants are prone to several diseases that can severely affect yield and quality. Common rice leaf diseases include:

**Bacterial Blight**: Caused by Xanthomonas oryzae, it results in water-soaked lesions that can spread rapidly (as shown in Figure 1.2).



**Figure 1.2: Bacterial Blight**

**Leaf Blast:** Caused by the Magnaporthe oryzae fungus, it leads to lesions on the rice leaves that hinder photosynthesis (as shown in Figure 1.3).



**Figure 1.3: Leaf Blast**

**Brown Spot:** A fungal infection caused by Bipolaris oryzae, which causes brown spots on rice leaves(as shown in Figure 1.4).



**Figure 1.4: Brown Spot**

**Tungro** : Transmitted by leafhoppers, this virus stunts plant growth and reduces yield. Due to the similarity in symptoms among different diseases, manual identification can be challenging, emphasizing the need for precise automated systems for early detection (as shown in Figure 1.5).



**Figure 1.5: Tungro**

**Hispa**: caused by the beetle Dicladispa armigera, primarily harms rice leaves by feeding on the outer layers, resulting in linear, white streaks. This damage weakens the plant, decreases photosynthesis, and can significantly reduce yield if not addressed quickly (as shown in Figure 1.6).



**Figure 1.6: Hispa**

**1.3 Limitations of Traditional Disease Detection Methods**

Traditional methods for detecting diseases, such as manual assessments by agricultural experts, can be labour-intensive, subjective, and often lead to errors. The symptoms of various diseases frequently overlap, making accurate diagnosis challenging. Additionally, delays in detection can result in significant crop losses. This underscores the necessity for automated and efficient disease detection systems that can swiftly and accurately identify diseases in rice plants.

## 1.4 Role of Artificial Intelligence in Disease Identification

Artificial Intelligence (AI) has revolutionized the agricultural industry, especially in identifying plant diseases. Deep learning models like DenseNet121 have demonstrated outstanding performance in classifying plant diseases. DenseNet121 employs a dense network architecture that allows for efficient feature reuse, enhancing the model’s capability to accurately classify rice leaf diseases. This method greatly diminishes the dependence on manual techniques, providing a faster and more precise disease detection process.

**1.5 Integrating AI and DMD for Improved Disease Detection**

The integration of DenseNet121 with DMD Preprocessing presents an innovative approach to detecting rice leaf diseases. While DenseNet121 focuses on accurate disease classification, DMD enhances region segmentation by highlighting infected areas and minimizing irrelevant background information. This combination improves the overall effectiveness and reliability of the detection system.

## 

## 1.6 Motivation

The integration of DenseNet121 and Dynamic Mode Decomposition (DMD) in our project for detecting rice leaf diseases tackles significant challenges in rice farming. Rice is a crucial staple crop that sustains millions of people globally, especially in Asia. Yet, it is highly vulnerable to diseases such as bacterial blight, leaf blast, and brown spots. These diseases can drastically reduce both yield and quality, making prompt and accurate detection essential to minimize losses. Conventional disease detection methods tend to be slow, labour-intensive, and error-prone, highlighting the necessity for an efficient, automated solution.

We selected DenseNet121 due to its outstanding performance in image classification, which allows for precise identification of diseases through its sophisticated architecture. Its capability to capture both low-level and high-level features makes it effective in differentiating between healthy and diseased rice leaves. By utilizing DenseNet121, our project aims to enhance prediction accuracy, providing farmers with a dependable tool for early diagnosis and intervention.

To boost performance further, we incorporate Dynamic Mode Decomposition (DMD) into the system for Preprocessing input data. DMD extracts essential spatial-temporal features, directing the model's focus on disease-affected areas while filtering out irrelevant background information. This method enhances the segmentation of infected regions, allowing the model to predict, classify, and localize rice leaf diseases accurately. This approach facilitates early disease detection, enabling farmers to manage affected areas more effectively, reduce crop losses, and improve their revenue.

# CHAPTER 2

# LITERATURE SURVEY

## 2.1 Overview

The literature survey offers a detailed examination of earlier research and developments in the area of plant disease detection, image Preprocessing, and classification models based on deep learning. The studies reviewed here investigate a range of methodologies, from conventional machine learning methods to sophisticated convolutional neural network (CNN) architectures. Additionally, it highlights the importance of Preprocessing techniques such as Dynamic Mode Decomposition (DMD) in improving the segmentation of disease regions and the accuracy of classification.

## 2.2 Literature Review

The table below summarizes key studies relevant to rice disease detection using deep learning models.

**Table 1: Literature Survey**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Title | Journal & Authors | Year | Techniques Used | Observation |
| Advancements in Rice Disease Detection Through Convolutional Neural Networks: A Comprehensive Review | Heliyon, Burak Gülmez | 2024 | Comparative analysis of CNN-based techniques including hyper parameter optimization | Models like GoogleNet achieved 92%, and others ranged between 90–94%. |
| Automatic Recognition of Rice Plant Leaf Diseases Detection Using Deep Neural Network with Improved Threshold Neural Network | e-Prime Advances in Electrical Engineering, K. Mahadevan, A. Punitha, J. Suresh | 2024 | DSGAN2 with Improved Threshold Neural Network, Social Spider Optimization (S2O-FCW) | DSGAN2 achieved lower false detection rates: 55.2% (APS-DCCNN), 50.4% (AlexNet), 49.5% (CNN). Improved disease detection accuracy |
| A High-Quality Rice Leaf Disease Image Data Augmentation Method Based on Dual GAN | Zhao Zhang, Quan Gao, Lirong Liu, Yun He, IEEE Access | 2023 | Dual GANs (WGAN-GP and Opt-Real-ESRGAN) for generating high-quality augmented images for disease recognition. | Improved accuracy by 4.57%, ResNet18 achieved 93.57%, and VGG11 achieved 94.1% after augmentation. |
| Comparative Analysis of Transfer Learning, LeafNet, and Modified LeafNet Models | Wassem I. A. E. Altabaji et al., IEEE Access | 2024 | Transfer learning using Xception and MobileNetV2 with modified LeafNet architecture. | Achieved the highest accuracy of 97.44% on Modified LeafNet, enhanced via hyper parameter tuning and data augmentation |
| Convolutional Neural Network in Rice Disease Recognition: Accuracy, Speed, and Lightweight | Hongwei Ning et al., Frontiers in Plant Science | 2023 | Ensemble learning and lightweight CNN approaches such as model pruning and knowledge distillation for efficient disease detection. | DenseNet121: 94.33%, AlexNet: 98.8%, InceptionResNetV2: highest accuracy, Mobile-Net (lightweight model): 94.85%.. |
| Dynamic Mode Decomposition-Based Salient Edge/Region Features for Content-Based Image Retrieval | Sikha O K, Soman K P,  Multimedia Tools and Applications; | 2022 | Salient Edge/Region Detection using DMD | Achieved 88.2% precision and 86.5% recall on the Wang’s dataset for image retrieval. |
| Multi-Resolution Dynamic Mode Decomposition-Based Salient Region Detection in Noisy Images | O. K. Sikha, K. P. Soman Signal, Image, and Video Processing; | 2021 | Multi-Resolution Dynamic Mode Decomposition (MRDMD) | Achieved 89%-94% precision in salient region detection, with robustness across varying noise levels. |
| MC-DMD: A Data-Driven Method for Blood Vessel Enhancement in Retinal Images Using Morphological Closing and Dynamic Mode Decomposition | Journal of King Saud University – Computer and Information Sciences; Suchithra Madathil, Soman Kutti Padannayil​ | 2022 | Morphological Closing, Dynamic Mode Decomposition (DMD) | Achieved 96.23% accuracy, 84.7% sensitivity, and 98.8% specificity in vessel segmentation tasks. |

## 2.3 Research Gap

#### Previous methods for detecting plant diseases have primarily depended on traditional image segmentation and convolutional neural networks (CNNs). These techniques often struggle with accuracy due to the constraints of small, non-diverse datasets and their inability to effectively concentrate on areas affected by disease. The research gap highlights the necessity for an efficient Preprocessing step that eliminates irrelevant noise and sharpens the focus on disease-affected regions. This study aims to fill this gap by employing Dynamic Mode Decomposition (DMD) to enhance segmentation accuracy, which in turn facilitates better detection of rice leaf diseases. By combining DMD with DenseNet121, the proposed method seeks to improve disease detection and classification, offering a more dependable solution for disease management.

## 2.4 Problem Statement

After defining the research gap, the problem statement is defined.

To design and develop a deep learning-based model for the accurate detection and identification of rice leaf diseases, using Dynamic Mode Decomposition (DMD) to enhance disease region identification and improve crop yield management for rice farmers.

## 2.5 Objectives

1. Develop a model to accurately detect and identify diseases affecting rice plants, offering rice farmers timely insights to take preventive actions.
2. Localize the affected area by the identifying diseases on rice leaves, allowing farmers to assess the extent of damage and plan to minimize yield loss.
3. Use Dynamic Mode Decomposition (DMD) to preprocess images, reducing irrelevant background noise and focusing the model on disease-affected areas to improve classification accuracy.

## CHAPTER 3

# REQUIREMENT ANALYSIS

In this chapter, we explore the vital elements of requirement analysis, with a particular emphasis on the necessary software and hardware prerequisites for the system. These key components are collectively known as system requirements.

## 3.1 Functional Requirements

Image Preprocessing: Enhance the quality of input rice leaf images and standardize features like resolution, brightness, and contrast to ensure consistent input for the disease detection model.

**Model Architecture**: Utilize a DenseNet121-based architecture for disease classification, taking advantage of the pre-trained model for effective feature extraction and precise classification of rice leaf diseases.

**Dynamic Mode Decomposition (DMD) Integration:** Integrate DMD for Preprocessing to emphasize infected areas while minimizing irrelevant background noise, thereby improving the accuracy of disease detection and sharpening the model's focus on affected regions.

**Performance Metrics**: Assess the performance of the disease detection model using metrics such as accuracy, precision, recall, and F1-score to guarantee accurate classification and dependable detection of disease-infected areas in rice leaves.

## 3.2. Non-Functional Requirements

**Computational Efficiency:** Optimize algorithms for computational efficiency, considering real-time processing requirements.

**Scalability:** Design algorithms that scale effectively with varying image dataset sizes. Ensure the system's ability to handle both small-scale and large-scale datasets.

**User Interface:** Ensure accessibility for users with varying levels of technical expertise.

**Robustness:** Implement error-handling mechanisms to address potential issues during algorithm execution.Conduct rigorous testing with diverse dataset images to ensure robust performance.

## 3.3 Hardware Components

Latest OS: Windows 11

Processor: 2 GHz (Giga Hertz)

RAM: 8GB (Giga Byte) for 32-bit or 8GB for 64-bit.

Hard disk space: 16 GB for 32-bit OS or 20 GB for 64-bit OS.

## 3.4 Software Components

**3.4.1 Google Colab**

### Google Colab serves as a cloud-based software component, offering a collaborative environment for developing and executing Python code, including machine learning and deep learning algorithms. With its integration of Jupyter notebooks, Colab provides a seamless platform for prototyping, training, and evaluating segmentation models, facilitating efficient collaboration and computation without the need for local hardware resources. Its accessibility, interactive features, and compatibility with popular libraries such as TensorFlow and PyTorch make it an ideal choice for researchers and practitioners in medical imaging.

### 3.4.2 Python Libraries

The Python libraries included here in our project are

* **OS**

The OS module in Python includes functions for adding and deleting directories (folders), retrieving their contents, altering and identifying the current directory, etc. To communicate with the underlying operating system, you must first import the OS module.

* **NumPy**

The open-source NumPy (Numerical Python) library is used with the Python programming language. It is employed in scientific computing and array manipulation. It offers high-level operating capabilities for working with arrays in addition to its multidimensional array object.

* **Cv2**

The Cv2 module processes images and videos to detect objects, people, or even human handwriting. The number of weapons in your arsenal rises when it is integrated with other libraries, such as NumPy, which is a highly optimised library for numerical operations. All operations that can be performed with NumPy can be combined with OpenCV.

* **TensorFlow**

High-performance numerical calculations can be performed using the open-source software library TensorFlow. Its adaptable architecture makes it simple to deploy computation across a range of platforms (CPUs, GPUs, and TPUs), from desktop computers to server clusters to mobile and edge devices.

* **Pytorch**

PyTorch is a popular open-source machine learning framework known for its flexibility and dynamic computation graphs, making it suitable for building and training neural network models. With GPU acceleration and automatic differentiation, PyTorch facilitates efficient training of complex models, including those used in medical image segmentation.

* **MatPlot**

Matplotlib is a feature-rich Python data visualization package that makes it easier to create interactive, animated, and static representations in a variety of formats. With the many plotting functions offered by this strong and adaptable library, users can create a variety of plots, including line charts, scatter plots, histograms, and more. Because of Matplotlib adaptability, customization possibilities, and smooth interaction with other Python libraries, it is extensively utilized in scientific computing, data analysis, and machine learning.

Matplotlib, which offers an extensive range of visualization tools, is essential for clearly and visually pleasingly communicating data-driven insights through plots.

This chapter covered the program's hardware and software requirements as well as the Python libraries that were utilized. The dataset description will be covered in the following chapter.

# CHAPTER 4

**DATASET DESCRIPTION**

This chapter provides an overview of the dataset used in this study, which is essential for training and evaluating the model aimed at detecting and identifying diseases in rice leaves. The dataset comprises 9,607 images obtained from Kaggle, showcasing a range of rice leaf conditions, including healthy leaves and various rice leaf diseases. By leveraging this diverse dataset, the model can be trained to accurately classify and identify diseases in rice plants, ultimately aiding rice farmers in improving yield management.

## 4.1 Data Overview

The dataset includes images of healthy rice leaves and five prevalent rice diseases: Blast Disease, Brown Spot, Bacterial Blight, Hispa, and Tungro. These images are the input for the deep learning model, enabling it to learn the unique features and characteristics associated with each category. Each image features a single rice leaf, and the model's task is to identify the disease present or classify the leaf as healthy. The dataset is organized into the following categories (as shown in Figure 4.1).

Healthy Rice Leaves: 1,488 images

Blast Disease: 2,299 images

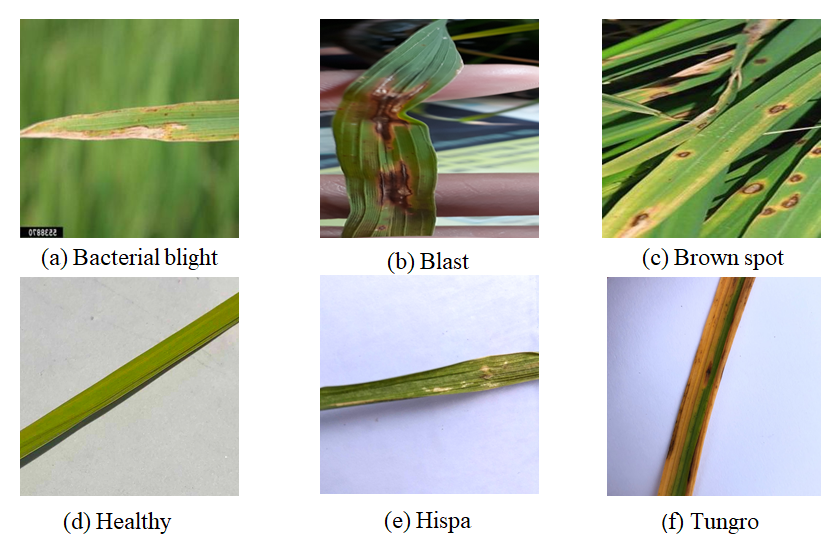
Brown Spot: 2,163 images

Bacterial Blight: 1,704 images

Hispa: 565 images

Tungro: 1,388 images

Each category represents a distinct class that the model must recognize. The images depict various conditions of rice leaves, ranging from healthy plants to those affected by diseases, with variations in size, angle, and lighting. This diversity in the dataset enhances the model's ability to generalize and effectively manage real-world image variations.

****The images of rice leaves showcase a wide range of characteristics, including the overall shape and texture of the leaves, as well as subtle indicators of disease such as discoloration, lesions, and spots. The model's effectiveness in detecting these features hinges on its ability to learn from this diverse collection of images. For example, Blast Disease may result in yellow or brown spots, while Brown Spot is known for its distinctive speckling patterns on the leaves. By analyzing these images, the model is anticipated to accurately identify and categorize the diseases.

**Figure 4.1: Sample Dataset Images**

## 4.2 Dataset Split

To train and test the model effectively, the dataset is divided into two main subsets: the training set and the testing set. An 80-20 split strategy was implemented for both the images and their corresponding labels across all classes, meaning that 80% of the dataset is allocated for training the model, while the remaining 20% is set aside for evaluating the model's performance. This method ensures that the model is assessed on unseen data, offering a reliable estimate of its ability to generalize.

**4.2.1 Training Set**

The training set consists of 80% of the total dataset. It features a balanced representation of all classes, enabling the model to learn the characteristics of both healthy and diseased rice leaves.

Total Images: 80% of 9607 = 7685 images

Healthy: 80% of 1488 = 1190 images

Blast Disease: 80% of 2299 = 1839 images

Brown Spot: 80% of 2163 = 1730 images

Bacterial Blight: 80% of 1704 = 1363 images

Hispa: 80% of 565 = 452 images

Tungro: 80% of 1388 = 1111 images

The training set includes images from all six categories, which are utilized to instruct the model on how to classify each disease as well as the healthy category. The model will learn to recognize key features linked to each disease while differentiating healthy leaves from those that are diseased. The training images undergo Preprocessing to ensure they maintain a consistent format, providing the model with high-quality data for effective learning.

**4.2.2 Testing Set**

The testing set, which accounts for 20% of the total dataset, includes images that were not part of the training process. These images are specifically set aside to evaluate the model's performance, ensuring it can generalize to new, unseen data. The testing set is categorized in the same way as the training set, allowing for a balanced assessment of the model's capabilities.

Total Images: 20% of 9607 = 1922 images

Healthy: 20% of 1488 = 298 images

Blast Disease: 20% of 2299 = 460 images

Brown Spot: 20% of 2163 = 433 images

Bacterial Blight: 20% of 1704 = 341 images

Hispa: 20% of 565 = 113 images

Tungro: 20% of 1388 = 277 images

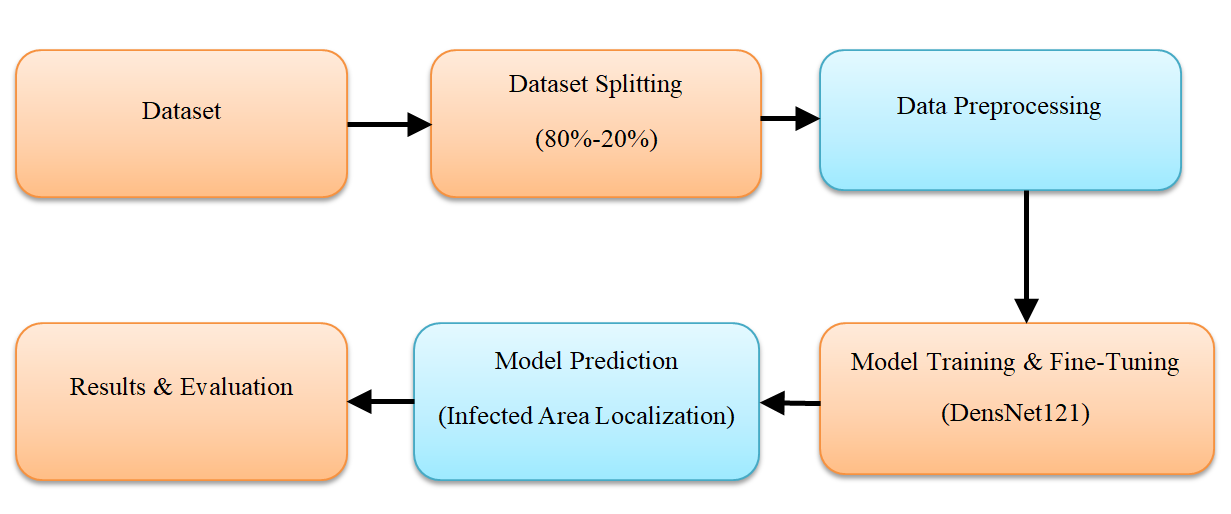
Testing images are utilized to evaluate the model’s accuracy, precision, recall, and F1-score, ensuring it can accurately identify the various diseases and distinguish them from healthy leaves. The testing set offers important insights into the model's performance in real-world scenarios, where the data may differ from what was used in training.

# CHAPTER 5

**METHODOLOGY**

## 5.1 Methodology Overview

The methodology chapter details the structured approach taken to create and assess the proposed system, AI-Based Rice Leaf Disease Identification Enhanced by Dynamic Mode Decomposition. It outlines the processes in preparing the dataset, Preprocessing with Dynamic Mode Decomposition (DMD), extracting features, segmenting the data, classifying using DenseNet121, and evaluating performance.

This chapter offers a comprehensive explanation of the system architecture, implementation specifics, and the techniques used to enhance segmentation and classification accuracy (as shown in Figure 5.1).

**Figure 5.1: High-Level Design**

The proposed system for detecting and classifying rice leaf diseases combines advanced techniques like Dynamic Mode Decomposition (DMD) and DenseNet121, which is a cutting-edge convolutional neural network (CNN). This methodology aims to provide robustness, scalability, and high accuracy in identifying rice diseases. In this section, we will outline the steps and components of the methodology, including dataset preparation, Preprocessing, feature extraction, model training, and evaluation.

## ****5.2 Data Collection and Dataset Preparation****

#### ****5.2.1 Data Collection****

The dataset used in this project comprises 9,607 labelled images of rice leaves, obtained solely from the open-access platform Kaggle. It is well-organized and includes images that represent six different categories of rice leaf health statuses. These categories are:

**Healthy:** Rice leaves that show no visible signs of disease or damage.

**Blast Disease:** Caused by Magnaporthe oryzae, this condition is marked by spindle-shaped lesions with a grey centre and a brown border.

**Brown Spot**: Resulting from Cochliobolus miyabeanus, this disease features circular or oval brown lesions.

**Bacterial Blight**: This bacterial infection, is caused by Xanthomonas oryzae pv. oryzae, is characterized by water-soaked streaks and wilting of the leaves.

**Hispa**: Damage from Dicladispa armigera leads to skeletonized patches on the leaves due to insect feeding.

**Tungro:** A viral infection that causes stunted growth and a yellow-orange discoloration of the leaves.

All images in the dataset are pre-labelled, providing high-quality annotations that are crucial for training a supervised learning model. The thorough labelling of the dataset allows the model to effectively distinguish between subtle symptoms of disease.

**5.2.2 Dataset Preparation**

To ensure compatibility with the DenseNet121 model and enhance the quality of the input data, several Preprocessing steps were implemented on the dataset. These steps include resizing, normalization, data splitting, augmentation, and noise reduction.

**5.2.3 Resizing**

All images were resized to a consistent resolution of 224 times 224 pixels, which is the standard input size for the DenseNet121 model. This resizing process ensures that the model can process all images uniformly, preventing computational inefficiencies and distortions that may arise from varying resolutions in the original dataset.

**5.2.4 Normalization**

Pixel intensity values were normalized to a range of [0, 1] by dividing each pixel value by 255, the maximum value for an 8-bit image. This normalization step standardizes the input data, enhancing the stability of the training process and speeding up convergence.

**5.2.5 Data Splitting**

The dataset was divided into two subsets using an 80-20 split:

Training Set (80%): This subset, consisting of 7685 images, was utilized to train the DenseNet121 model.

Testing Set (20%): This subset, containing 1922 images, was set aside for evaluating the model’s performance on unseen data. This division ensures that a substantial portion of the data is available for training the model, while the remaining subset allows for an unbiased assessment of the model’s generalization ability.

**5.2.6 Data Augmentation**

To address the imbalance in class distribution, data augmentation techniques were employed to enhance the representation of minority classes, such as Hispa. The augmentation techniques used include:

**Random Rotations:** Up to ±30 degrees, simulating various orientations of leaves.

**Flipping**: Both horizontal and vertical flips to increase variability.

**Brightness Adjustments:** Modifying the brightness to replicate different lighting conditions.

**Scaling and Zooming:** Randomly zooming into the image to emphasize different areas of the leaf.

Data augmentation enhances the model's robustness by exposing it to a wider range of image transformations.

## 5.3 Dynamic Mode Decomposition (DMD) For Preprocessing

**5.3.1 Introduction to DMD Dynamic Mode Decomposition (DMD)**

Dynamic Mode Decomposition (DMD) is a data-driven algorithm that was initially developed in fluid mechanics to study how fluid flows change over time. Its application has since broadened to other fields, including image processing, thanks to its effectiveness in managing high-dimensional and nonlinear systems. A key benefit of DMD is that it does not require explicit governing equations, allowing it to extract significant spatial-temporal structures from datasets.

In this project, DMD is employed as a Preprocessing technique to identify and emphasize disease-affected areas in rice leaf images. By breaking down images into low-rank (static) and sparse (dynamic) components, DMD successfully minimizes irrelevant background details and highlights areas of interest, such as lesions and discolorations that signal diseases. This Preprocessing step improves the subsequent classification model's ability to concentrate on essential features, enhancing accuracy and interpretability. The sparse component generated by DMD acts as a hard attention map, directing the model's focus toward disease-specific areas while minimizing noise.

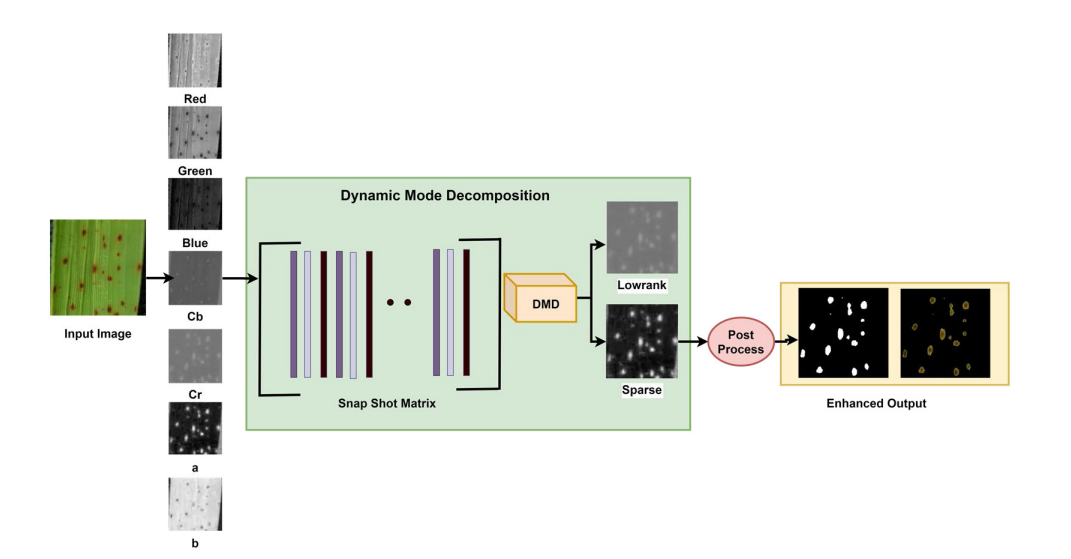
The mathematical basis of DMD approximates nonlinear systems by calculating the eigenvalues and eigenvectors of a linear Koopman operator. This operator captures the dynamics between consecutive observations, effectively representing both static and dynamic aspects of the data (as shown in Fig 5.2)

**Benefits of DMD**

1. **Noise Reduction** DMD effectively separates disease-relevant features from static background elements, such as soil and overlapping leaves. This improves the clarity and relevance of the input images.

2. **Enhanced Focus on Disease Regions** By highlighting dynamic components, DMD allows the classifier to concentrate on areas with disease-specific features, enhancing the model’s performance and accuracy.

3. **Improved Interpretability** The segmented images produced through DMD offer clear visual representations of disease-affected areas. This enhances interpretability and enables stakeholders to visually validate the model’s predictions.

****

**Figure 5. The DMD preprocessing architecture**

**Figure 5.2: The DMD Preprocessing architecture.**

* + 1. **Mathematical Representation of DMD**

**Snapshot Matrix Construction:**

In the context of a dataset, a snapshot matrix captures observations (such as images) at various states.

()

Each as shown in Equation (1).

**Shifted Snapshot Matrices:**

Two matrices, X1 and X2, are constructed to Model temporal relationships:

()

The matrices and represent the original and time-shifted observations, respectively, as shown in Equation (2).

**Koopman Operator Representation:**

The relationship between ​ is modelled as

()

Here, A represents the **Koopman operator**, which captures the dynamics of the system, as shown in Equation (3).

**Computation of Koopman Operator:**

To compute A, the Moore-Penrose pseudo inverse is used:

()

The pseudo inverse X1 is calculated as:

()

As shown in Equations (4) and (5), the Koopman operator is derived by using the pseudo-inverse of **.**

**Singular Value Decomposition (SVD):**

using SVD:

()

where:

The decomposition of using SVD is shown in Equation (6)

**Reduced Koopman Operator**

()

As shown in Equation (7), the reduced Koopman operator is computed to capture the dynamics efficiently.

**Dynamic Modes and Sparse Component:**  
The Eigen decomposition of yields:

()

where W contains eigenvectors, represents dynamic modes highlighting disease regions.

Equation (8) provides the dynamic modes extracted using and

**5.3.3 Implementation Steps**

**Step 1**: Input Image Analysis Each image is treated as a snapshot matrix, where pixel intensities serve as observations. For colour images, the RGB channels are transformed into alternative colour spaces like CIE L\*a\*b, YCbCr, and YUV. These transformations take advantage of variations in intensity and chrominance, as diseased areas typically show noticeable colour differences.

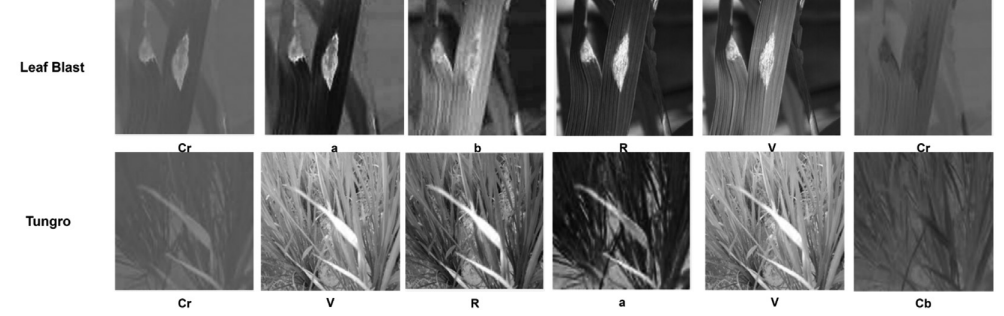
**Step 2**: Sparse Component Extraction Utilizing the Koopman operator, DMD breaks down the snapshot matrix into:

Low-Rank Component: This represents static background features, including soil and healthy leaf regions.

Sparse Component : This captures dynamic changes related to lesions, discoloration, or other disease symptoms. The sparse component helps isolate areas of interest by filtering out irrelevant background details.

**Step 3:** Attention Map Generation The sparse component is processed to create a binary mask known as an attention map. Thresholding is applied to keep regions with significant variability, which correspond to disease-affected areas.

**Step 4:** Segmented Image Creation The attention map is superimposed on the original image to produce a segmented output. This segmented image highlights the regions affected by disease, directing the classifier to concentrate on relevant features during the learning process.



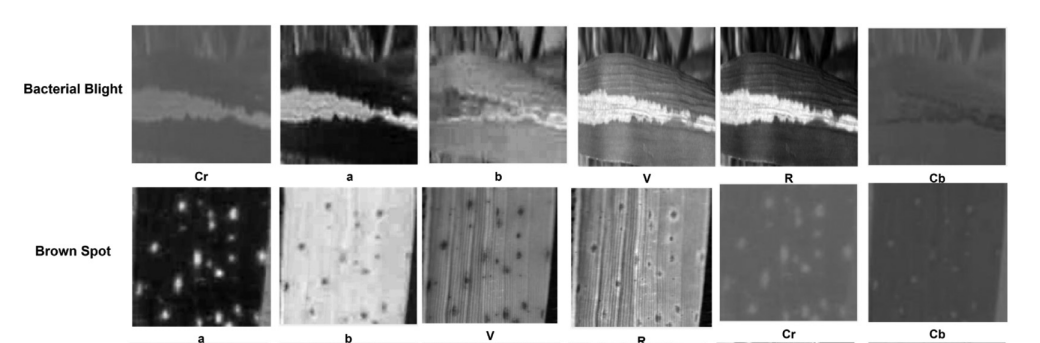
****

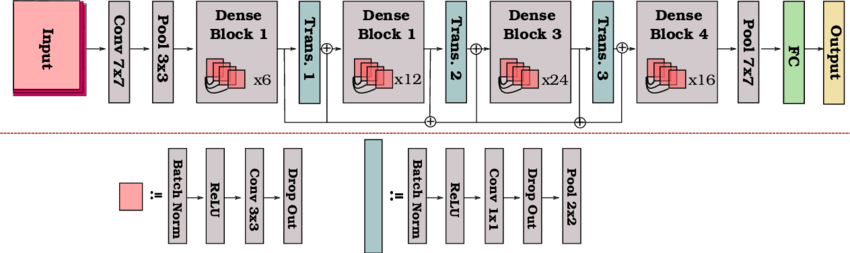
Fig 7. Color channel representation of rice leaf images

**Figure 5.3 :Color channel representation of rice leaf images**

## 5.4. Densenet121 For Feature Extraction And Classification

**5.4.1 DenseNet121 Architecture**

DenseNet121 is a deep learning model that utilizes dense connections between layers to improve gradient flow and minimize redundancy. This architecture is well-suited for fine-grained classification tasks like rice disease detection.

****The DenseNet121 architecture is particularly effective for classifying rice leaf diseases due to its innovative approach to feature reuse through dense connections. This section highlights its main components and their contributions to the classification task.

**Figure 8:Architecture of DensNet**

**Figure 5.4 : DenseNet121 Architecture**

**1 Dense Blocks**

At the core of DenseNet121 are dense blocks, which consist of groups of convolutional layers where each layer receives input from all previous layers within the block. This design promotes:

**Efficient Feature Propagation:** Each layer can access features learned earlier, minimizing redundancy.

**Enhanced Gradient Flow:** This structure supports smooth back propagation, reducing the likelihood of vanishing gradients in deeper networks. Each dense block includes:

**Batch Normalization (BN):** This process normalizes feature maps, speeding up training and

enhancing convergence.

**ReLU Activation:** This adds non-linearity, allowing the model to capture complex patterns.

**Convolutional Layers:** These layers apply filters to extract spatial features.

**2 Transition Layers**

Transition layers are placed between dense blocks to manage the model’s complexity and control feature dimensions. They serve two main functions:

Dimensionality Reduction: They use convolutions to decrease the number of feature maps, optimizing memory and computational efficiency. Down sampling: Average pooling is applied to lower the spatial resolution, retaining crucial information while eliminating redundancies.

**3 Global Average Pooling (GAP)**

DenseNet121 uses Global Average Pooling in place of traditional fully connected layers, which provides: Reduced Over fitting: This approach removes the need for millions of parameters typically associated with dense layers.

Improved Interpretability: It aggregates spatial information, creating a clear link between feature maps and class probabilities.

**4 Softmax Classifier**

The final output layer is a softmax classifier that calculates probabilities for each of the six rice leaf disease categories:

Healthy

Blast Disease

Brown Spot

Bacterial Blight

Hispa

Tungro

The classifier assigns probabilities accordingly.

**5.4.2 Fine-tuning DenseNet121**

means adjusting the pre-trained model specifically for detecting rice leaf diseases using transfer learning. This process allows the model to utilize previously learned features while focusing on the new task.

**1 Pre-Trained Weights**

DenseNet121 starts with weights that were pre-trained on the ImageNet dataset. These weights offer:

Generic Feature Extraction: The model benefits from pre-learned features such as edges, textures, and shapes, which serve as a base for more specific learning.

Reduced Training Time: Using pre-trained weights helps speed up the training process.

**2 Freezing Initial Layers**

During the initial training phase, the early convolutional layers, which capture basic features, were kept frozen to maintain their general-purpose knowledge. This strategy:

Helps prevent over fitting by limiting the number of trainable parameters. Preserves strong features that can be applied across different datasets.

**3 Adding Custom Layers**

To tailor the pre-trained DenseNet121 architecture for rice leaf disease classification, custom layers were added. These layers included:

Fully Connected Layer: This layer reduces the feature dimensions to align with the six disease classes.

Dropout Layer: This was introduced to help prevent over fitting by randomly turning off neurons during the training process.

Softmax Activation: This layer transforms the outputs into probabilities for each class.

**4 Unfreezing and Fine-Tuning**

Once the custom layers were trained, the last few layers of DenseNet121 were unfrozen for fine-tuning.

This process involved: Training with a lower learning rate to fine-tune domain-specific weights while preserving the knowledge from the pre-trained model.

Optimizing features that highlight subtle differences in lesion shapes, textures, and colors that are characteristic of rice leaf diseases.

**5.4.3 Training Workflow**

**Data Augmentation:** Techniques like rotation, flipping, and brightness adjustments were used to enhance model generalization and minimize over fitting.

**Optimizer:** The Adam optimizer was chosen for its adaptive learning capabilities, along with a learning rate scheduler to gradually decrease the learning rate.

**Loss Function:** Categorical cross-entropy was used to evaluate the error in multi-class predictions.

Early Stopping: Training was stopped when the validation loss ceased to improve, which helped to avoid over fitting.

## Advantages of DenseNet121

**Efficiency:** The dense connections in the model enhance feature reuse, leading to a more compact and efficient representation.

**Compact Design**: With fewer parameters compared to other architectures, this model is computationally efficient.

**Generalization:** Utilizing pre-trained weights and fine-tuning helps achieve strong performance on a variety of unseen data.

# CHAPTER 6

**IMPLEMENTATION**

This chapter discusses how the proposed Model, AI-Based Rice Leaf Disease Identification Enhanced by Dynamic Mode Decomposition (DMD), is implemented. The system utilizes the DenseNet121 model for classification and incorporates DMD to accurately identify infected areas. The upcoming sections will offer a comprehensive overview of dataset preparation, Preprocessing, model training, and evaluation**.**

## 6.1 Data Collection And Preprocessing

The dataset for this project consists of 9,607 labelled images of rice leaves, divided into six categories: Healthy, Blast Disease, Brown Spot, Bacterial Blight, Hispa, and Tungro. Each category corresponds to distinct symptoms, including spindle-shaped lesions, circular spots, water-soaked streaks, skeletonized patches, and discoloration. The dataset underwent Preprocessing to prepare it for both training and evaluation. The Preprocessing steps involved resizing all images to 224x224 pixels to meet the input size requirements of DenseNet121, as well as normalizing pixel values to a range of [0, 1] for uniformity. The dataset was divided into training (80%) and testing (20%) subsets. To tackle class imbalance and improve generalization, data augmentation techniques such as rotation, flipping, brightness adjustment, and zooming were employed to generate diverse representations of the images.

## ****6.2 Dynamic Mode Decomposition (DMD) Preprocessing****

To improve the model's focus on disease-specific features, **Dynamic Mode Decomposition (DMD)** was employed. DMD decomposed the images into:

**Low-Rank Component**: Representing static background elements.

**Sparse Component**: Highlighting dynamic regions, such as lesions or discolorations.

The sparse component was processed to generate **binary attention maps**, isolating the infected areas. These maps were overlaid on the original images to create segmented outputs, effectively guiding the model to concentrate on disease-affected regions during training.

### 

### 

### 

**Figure 6.1: Sample rice leaf images and the results from DMD based preprocessing**

## 

## ****6.3 Densenet121 Model Implementation****

**DenseNet121,** a convolutional neural network, was chosen for its efficiency and accuracy in classification tasks. Its architecture leverages:

**Dense Connectivity**: Facilitating feature reuse and improved gradient flow.

**Transition Layers**: Reducing feature dimensions to optimize computational efficiency.

**Global Average Pooling (GAP)**: Preventing over fitting by summarizing spatial features.

**Softmax Classifier**: Assigning probabilities to each of the six disease classes.

**Transfer learning** was utilized by initializing the DenseNet121 model with pre-trained **ImageNet weights.** The early layers, capturing generic features like edges, were frozen during initial training. Custom layers, including fully connected layers, dropout layers for regularization, and a softmax activation layer, were added for disease classification. Finally, the last few layers of DenseNet121 were unfrozen and fine-tuned with a low learning rate to enhance the model's ability to detect disease-specific patterns.

## ****6.4 Framework And Libraries****

The implementation was carried out using Python and its extensive library ecosystem:

**PyTorch:** was chosen as the main deep learning framework due to its flexibility, dynamic computation graph, and robust support for research tasks, especially in training the DenseNet121 model.

**NumPy:** was utilized for efficient numerical operations and array manipulations, which allowed for smooth data handling and Preprocessing.

**OpenCV:** was essential for image Preprocessing, including tasks like resizing, normalization, and augmentation techniques such as rotation, flipping, and brightness adjustments.

**Matplotlib:** was used to visualize data distributions, model performance metrics, and segmented images from the DMD process. Together, these libraries enabled the development, training, and evaluation of the system, ensuring effective Preprocessing, precise model training, and clarity in the results.

## 6.5 Training the Model

The training process for the rice leaf disease classification model was meticulously crafted to ensure high accuracy while also being computationally efficient. The model utilized DenseNet121, a pre-trained convolutional neural network (CNN) architecture, to effectively classify rice leaf diseases. The training parameters and methodology were carefully chosen to align with the dataset and task requirements, guaranteeing dependable performance.

**6.5.1 Training Parameters**

**Epochs**

The model underwent training for 10 epochs, which proved to be adequate for the dataset and task at hand. This choice was informed by the observation that the model converged well within this range, achieving high accuracy without succumbing to over fitting.

**Optimizer**

The Adam optimizer was implemented, featuring a learning rate of 0.0001.

Adam is well-regarded for its adaptive learning rate mechanism, which merges the advantages of momentum and RMSProp optimizers. The chosen learning rate struck a balance between stable convergence and effective learning.

**Batch Size**

A batch size of 32 was selected, enabling the model to handle a reasonable number of images at once without straining memory resources. This size is optimal for stabilizing gradient updates and ensuring computational efficiency.

**Loss Function**

The Categorical Cross-Entropy loss function was utilized, as it is particularly suited for multi-class classification tasks. This function measures the disparity between the predicted probabilities and the actual class labels, guiding the model to reduce classification errors throughout the training process.

**6.5.2 Training Process**

The training process for the DenseNet121 model aimed at classifying rice leaf diseases involved four key steps: initialization, forward pass, backward pass, and evaluation. Each of these steps was essential for ensuring the model's accuracy and reliability.

**1. Initialization**

The model was initialized with pre-trained DenseNet121 weights sourced from the ImageNet dataset. This transfer learning strategy enabled the model to leverage previously learned feature extraction capabilities, concentrating on recognizing disease-specific patterns in rice leaf images, including symptoms of Bacterial Blight, Blast Disease, Brown Spot, Hispa, and Tungro.

**2. Forward Pass**

Input images were resized and preprocessed to align with the requirements of the DenseNet121 architecture. During the forward pass, these images were fed through the network's layers to extract features and calculate class probabilities for each type of disease. These probabilities reflected the

model's predictions regarding the likelihood of an image belonging to a particular class.

**3. Backward Pass**

The Categorical Cross-Entropy loss function was used to determine the error between the model's predicted probabilities and the actual labels. Through back propagation, the gradients of the loss concerning the model's weights were calculated. The Adam optimizer, with a learning rate set at

0.0001, iteratively updated the model’s weights, aiming to minimize the loss and enhance classification accuracy.

## 6.6 Evaluation

At the end of each epoch, the model was assessed using a separate validation dataset to track its performance. Metrics such as validation accuracy and validation loss were calculated to ensure the model could generalize effectively to new, unseen data. The model achieved a validation accuracy of 92.50%, demonstrating its capability to differentiate between similar disease classes.

**6.6.1 Evaluation Metrics**

The following metrics were utilized to evaluate the model's performance:

**1. Accuracy**

Accuracy indicates how correct the model’s predictions are overall. It is calculated as the ratio of correctly classified samples to the total number of samples as shown in **Equation 9**.

()

Where: TP: True Positives (correctly identified disease cases)

TN: True Negatives (correctly identified healthy cases)

FP: False Positives (healthy cases misclassified as disease)

FN: False Negatives (disease cases misclassified as healthy)

**2. Precision**

Precision measures the ratio of correctly predicted positive cases to the total predicted positive cases. Precision reflects the model's ability to minimize false positives, ensuring accurate disease identification as shown in **Equation 10**.

()

3**. Recall (Sensitivity)**

Recall assesses the ratio of correctly identified positive cases to the total actual positive cases. Recall gauges the model's effectiveness in identifying disease cases, aiming to reduce missed diagnoses as shown in **Equation 11**.

()

**4. F1 Score**

The F1 Score is the harmonic mean of Precision and Recall, offering a balanced assessment of the model’s performance. This metric is particularly valuable for datasets with imbalanced class distributions as shown in **Equation 12**.

()

# CHAPTER 7

# RESULTS AND ANALYSIS

The results and analysis section provides a comprehensive overview of the training and testing outcomes of the rice leaf disease classification model, utilizing the DenseNet121 architecture combined with Dynamic Mode Decomposition (DMD). The model's performance was evaluated using various metrics to assess its accuracy and effectiveness in classifying rice leaf diseases such as Blast Disease, Brown Spot, Bacterial Blight, Hispa, and Tungro, as well as its capability to localize infected regions on rice leaves.

The proposed work is based on DenseNet121 and incorporates Dynamic Mode Decomposition (DMD), was assessed using several metrics such as accuracy, precision, recall, and F1-score. The findings confirm the model's effectiveness in classifying rice leaf diseases and localizing infected areas.

## 7.1 Dynamic Mode Decomposition (DMD) Results

DMD Preprocessing The DMD algorithm successfully identified infected areas by isolating the sparse components that indicate disease-affected regions from the low-rank background representing healthy parts of the leaf. Visual results demonstrated DMD's capability to concentrate on regions that change dynamically, thereby improving the features specific to the disease for classification as shown in figure 7.1.

## 7.2 Key Observations

**Classification Performance**:

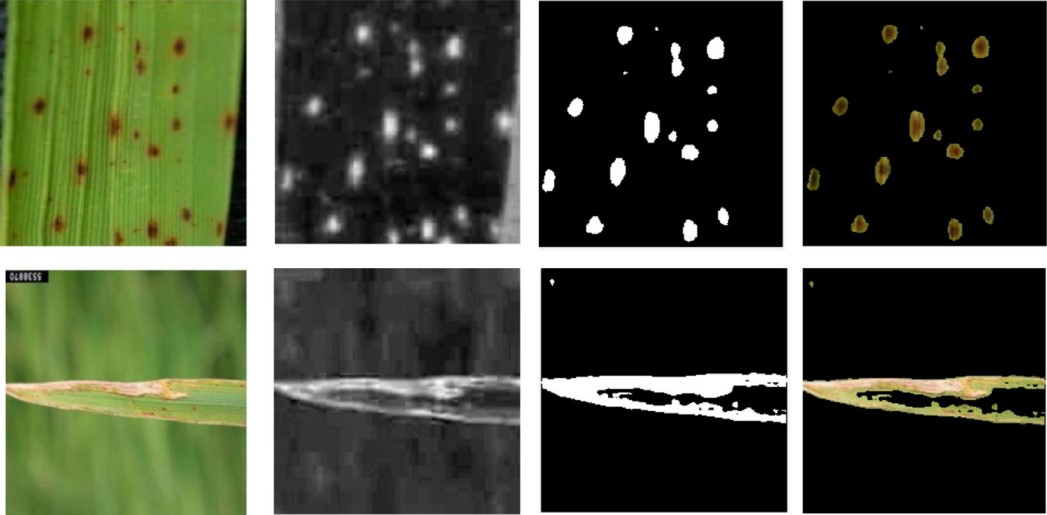
The model demonstrated nearly flawless precision and recall for diseases like Bacterial Blight and Tungro, showcasing the strength in managing unique disease characteristics. The slightly reduced metrics for Hispa can be explained by its symptom overlap with other classes.

**Infected Area Localization:**

The addition of DMD greatly enhanced the model’s interpretability by offering clear segmentation of infected areas. This provides valuable insights for farmers and agricultural specialists.

**Efficiency:**

The training process was completed efficiently in just 10 epochs, highlighting the model’s capability to converge quickly without falling into over fitting.



**Figure 7.1: Results after Dynamic Mode Decomposition**

## 7.3 Quantitative Results

#### ****7.3.1**** ****Model Performance****

#### The model was trained and achieved the following performance metrics:

* **Final Training Accuracy**: 93.32%
* **Final Training Loss**: 0.1964
* **Final Validation Accuracy**: 92.50%
* **Final Validation Loss**: 0.2089
* **Test Accuracy**: 91.59%

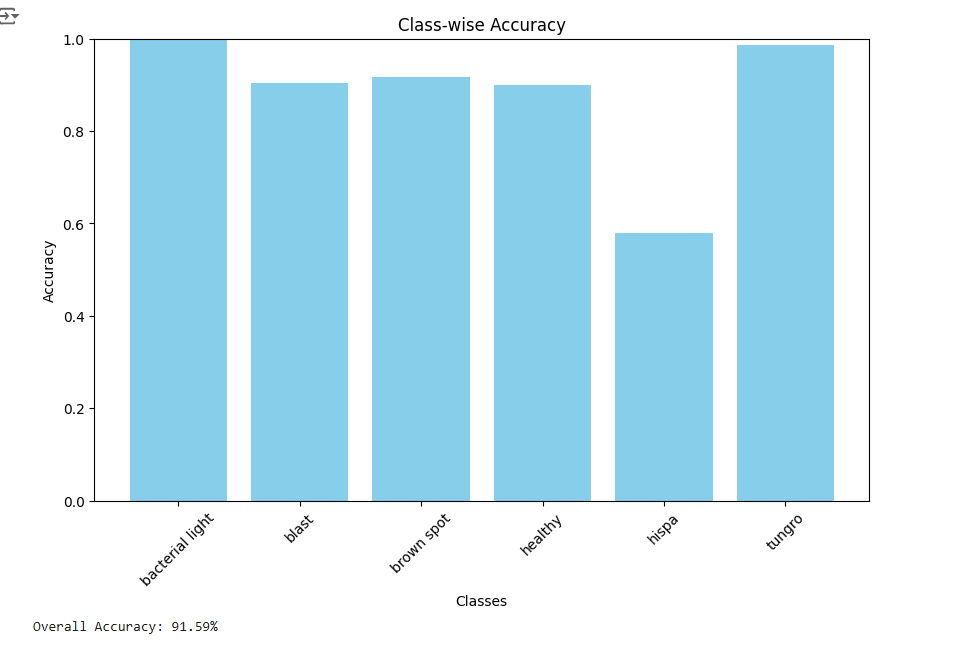
#### ****7.3.2 Classification Metrics****

* The model demonstrated strong performance across all disease classes:
* **Overall Test Accuracy**: 91.59% as shown in Table 7.3.2.

**Class-Specific Metrics Table 2** :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Precision** | **Recall** | **F1-Score** | **Support** |
| Bacterial Blight | 0.99 | 1.00 | 1.00 | 171 |
| Blast | 0.91 | 0.90 | 0.91 | 230 |
| Brown Spot | 0.97 | 0.92 | 0.94 | 217 |
| Healthy | 0.78 | 0.90 | 0.84 | 149 |
| Hispa | 0.67 | 0.58 | 0.62 | 57 |
| Tungro | 1.00 | 0.99 | 0.99 | 139 |
| **Overall Accuracy** |  |  | 0.92 |  |
| Macro Avg | 0.89 | 0.88 | 0.88 | 963 |
| Weighted Avg | 0.92 | 0.92 | 0.92 | 963 |

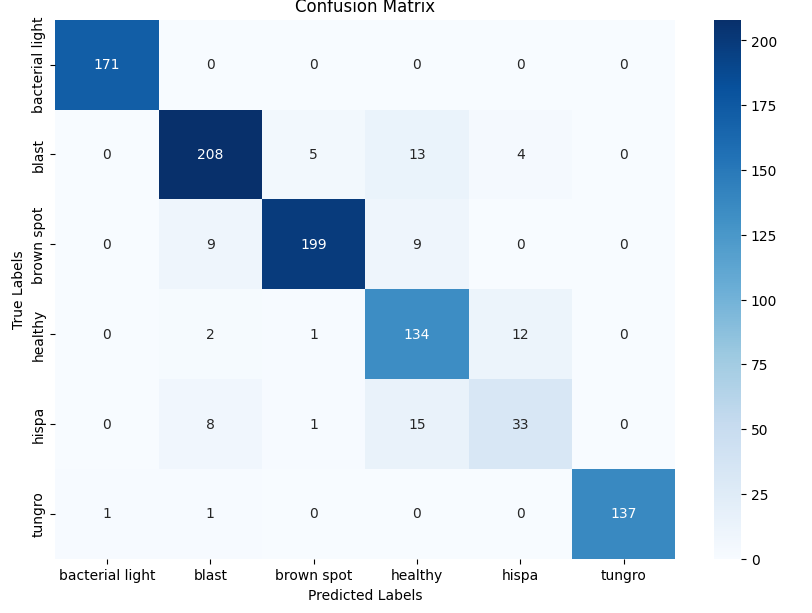
* **Macro Average**: Precision = 0.89, Recall = 0.88, F1-Score = 0.88
* **Weighted Average**: Precision = 0.92, Recall = 0.92, F1-Score = 0.92

****

**Fig 7.2:Graph Representing Accuracy for all the Classes**

**7.3.3 Confusion Matrix**

A confusion matrix was created to illustrate the model’s classification performance, displaying the true labels compared to the predicted labels for each class. The strong diagonal values suggest that the predictions for most classes are quite accurate.



**Figure 7.3: Confusion Matrix for all the Diseases**

## 7.4. Analysis of the Results

The results from the DenseNet121-based framework, combined with Dynamic Mode Decomposition (DMD), highlight its effectiveness in classifying rice leaf diseases and pinpointing infected areas. Achieving a test accuracy of 91.59%, the model performs well in differentiating diseases such as Bacterial Blight, Blast Disease, Brown Spot, Hispa, and Tungro. The high precision and recall scores for Bacterial Blight (1.00 for both) and Tungro (1.00 precision, 0.99 recall) demonstrate the model’s reliability in identifying specific disease patterns. However, the lower precision (0.67) and recall (0.58) for Hispa suggest that overlapping symptoms and visual similarities with other diseases present a challenge, highlighting the need for further refinement or augmentation of the dataset for this category. The overall macro-average precision of 0.89 and F1-score of 0.88 indicate that the model maintains balanced performance across all classes, addressing the imbalances in disease prevalence within the dataset. The incorporation of DMD greatly improved interpretability by isolating infected regions, allowing for clear visualization of dynamically changing areas linked to disease. This Preprocessing step effectively reduced background noise, enabling the model to concentrate on disease-affected regions, as shown by the sparse components and segmentation maps. The visual outputs, including segmentation maps overlaid on original images, validate the system’s capability to provide actionable insights for disease management. Additionally, the model converged within 10 epochs, with final training and validation losses of 0.1964 and 0.2089, respectively, showcasing its computational efficiency and stability. These results emphasize the potential of the proposed framework as a dependable and interpretable solution for early disease detection, enhancing agricultural productivity and reducing yield losses.

# CHAPTER 8

# CONCLUSION AND FUTURE SCOPE

The proposed AI-based model for identifying rice leaf diseases, enhanced with Dynamic Mode Decomposition (DMD), tackles challenges such as noisy backgrounds and class imbalances in agricultural diagnostics. By isolating disease-specific areas through DMD Preprocessing, the system boosts both classification accuracy and interpretability. DenseNet121, used with transfer learning, showed impressive feature extraction capabilities, achieving a training accuracy of 93.32%, validation accuracy of 92.50%, and test accuracy of 91.59%. The model successfully classified six types of rice leaf diseases, with conditions like Bacterial Blight and Tungro exhibiting high precision and recall. The segmentation maps generated by DMD offered valuable insights, making the system a powerful tool for early disease detection and intervention.

The combination of DMD Preprocessing with DenseNet121 highlights the potential of merging mathematical frameworks and deep learning to enhance agricultural diagnostics. This approach lowered misclassification rates by directing the classifier's focus to relevant areas, addressing the shortcomings of traditional CNN models. The segmentation maps created during Preprocessing provided a clear view of infected regions, allowing farmers to evaluate the severity of infections and respond promptly. The application of transfer learning significantly shortened training time while ensuring strong generalization across various data samples, showcasing the model’s practical usefulness.

Future advancements should aim to refine the DMD technique for even more precise segmentation and broaden the dataset to encompass additional disease categories and environmental factors. Investigating ensemble methods and integrating traditional machine learning with deep learning could further improve model accuracy and robustness. Moreover, adapting lightweight models for real-time application in the field and on mobile devices would enable farmers to receive immediate diagnostic results and recommendations. Expanding this approach would greatly benefit agricultural practices.

# REFERENCES

1. Burak Gülmez, "Advancements in Rice Disease Detection Through Convolutional Neural Networks: A Comprehensive Review," *Heliyon*, vol. 10, 2024, Art. no. e33328, doi: 10.1016/j.heliyon.2024.e33328.
2. K. Mahadevan, A. Punitha, and J. Suresh, "Automatic Recognition of Rice Plant Leaf Diseases Detection Using Deep Neural Network with Improved Threshold Neural Network," \*e-Prime - Advances in Electrical Engineering, Electronics and Energy\*, vol. 8, 2024, Art. no. 100534, doi: 10.1016/j.prime.2024.100534.
3. Zhao Zhang, Quan Gao, Lirong Liu, and Yun He, "A High-Quality Rice Leaf Disease Image Data Augmentation Method Based on Dual GAN," IEEE Access, vol. 11, 2023, pp. 21176–21188, doi: 10.1109/ACCESS.2023.3251098.
4. Wassem I. A. E. Altabaji, Muhammad Umair, Wooi-Haw Tan, Yee-Loo Foo, and Chee-Pun Ooi, "Comparative Analysis of Transfer Learning, LeafNet, and Modified LeafNet Models for Accurate Rice Leaf Diseases Classification," IEEE Access, vol. 12, 2024, pp. 36622–36632, doi: 10.1109/ACCESS.2024.3373000.
5. Hongwei Ning, Sheng Liu, Qifei Zhu, and Teng Zhou, "Convolutional Neural Network in Rice Disease Recognition: Accuracy, Speed, and Lightweight," Frontiers in Plant Science, vol. 14, 2023, Art. no. 1269371, doi: 10.3389/fpls.2023.1269371.
6. Nengyuan Pan, Weiming Yang, Yuting Luo, and Yonglin Wang, "Identification of Leaf Disease Based on Memristor Convolutional Neural Networks," *IEEE Access*, vol. 12, 2024, pp. 115197-115202, doi: 10.1109/ACCESS.2024.3444796.
7. Mehedi Hasan Bijoy, Nirob Hasan, Mithun Biswas, Suvodeep Mazumdar, Andrea Jimenez, Faisal Ahmed, Mirza Rasheduzzaman, and Sifat Momen, "Towards Sustainable Agriculture: A Novel Approach for Rice Leaf Disease Detection Using dCNN and Enhanced Dataset," *IEEE Access*, vol. 12, 2024, pp. 34174-34179, doi: 10.1109/ACCESS.2024.3371511.
8. G. K. V. L. Udayananda, C. Shyalika, and P. P. N. V. Kumara, "Rice Plant Disease Diagnosing Using Machine Learning Techniques: A Comprehensive Review," *SN Applied Sciences*, vol. 4, no. 311, 2022, doi: 10.1007/s42452-022-05194-7​(s42452-022-05194-7).
9. G. Latif, S. E. Abdelhamid, R. E. Mallouhy, J. Alghazo, and Z. A. Kazimi, "Deep Learning Utilization in Agriculture: Detection of Rice Plant Diseases Using an Improved CNN Model," *Plants*, vol. 11, no. 2230, pp. 1–17, 2022, doi: 10.3390/plants11172230​(plants-11-02230).
10. Y. K. Rathore, R. R. Janghel, S. K. Pandey, A. Kumar, K. U. Singh, and M. A. Shah, "A Novel AI-Based Approach for Better Segmentation of the Fungal and Bacterial Leaf Diseases of Rice Plant," *Journal of Sensors*, vol. 2022, Art. ID 6871085, pp. 1–12, 2022, doi: 10.1155/2022/6871085​(Journal of Sensors - 20).
11. K. Mahadevan, A. Punitha, and J. Suresh, "Automatic recognition of Rice Plant leaf diseases detection using deep neural network with improved threshold neural network," *e-Prime - Advances in Electrical Engineering, Electronics and Energy*, vol. 8, Art. ID 100534, pp. 1-15, 2024, doi: 10.1016/j.prime.2024.100534.
12. Latif, S. E. Abdelhamid, R. E. Mallouhy, J. Alghazo, and Z. A. Kazimi, "Deep Learning Utilization in Agriculture: Detection of Rice Plant Diseases Using an Improved CNN Model," *Plants*, vol. 11, Art. ID 2230, pp. 1-17, 2022, doi: 10.3390/plants11172230.
13. W. I. A. E. Altabaji, M. Umair, W.-H. Tan, Y.-L. Foo, and C.-P. Ooi, "Comparative Analysis of Transfer Learning, LeafNet, and Modified LeafNet Models for Accurate Rice Leaf Diseases Classification," *IEEE Access*, vol. 12, pp. 36622-36630, 2024, doi: 10.1109/ACCESS.2024.3373000.
14. Quan Gao, Lirong Liu, and Yun He, "A High-Quality Rice Leaf Disease Image Data Augmentation Method Based on a Dual GAN," *IEEE Access*, vol. 11, 2023, pp. 21176-21188, doi: 10.1109/ACCESS.2023.3251098.
15. Suchithra Madathil, Soman Kutti Padannayil, "MC-DMD: A Data-Driven Method for Blood Vessel Enhancement in Retinal Images Using Morphological Closing and Dynamic Mode Decomposition," Journal of King Saud University – Computer and Information Sciences, vol. 34, 2022, pp. 5223–5239, doi: [10.1016/j.jksuci.2022.06.002](https://doi.org/10.1016/j.jksuci.2022.06.002).
16. Sikha O K, Soman K P, "Dynamic Mode Decomposition-Based Salient Edge/Region Features for Content-Based Image Retrieval," Multimedia Tools and Applications, vol. 80, 2021, pp. 173–195, doi: 10.1007/s11042-020-10315-8.
17. O. K. Sikha, K. P. Soman, "Multi-Resolution Dynamic Mode Decomposition-Based Salient Region Detection in Noisy Images," Signal, Image and Video Processing, vol. 14, 2020, pp. 123–140, doi: [10.1007/s11760-019-01539-9](https://doi.org/10.1007/s11760-019-01539-9).