An Intelligent Disease Diagnostic Framework on Rice Plant Using Machine Learning

Final report of the major project is being submitted for fulfillment of the requirements for the degree of

Bachelor of Computer Application (BCA)

DISHA BANERJEE
Registration No: 223771010012 OF 2022-23
University Roll No: 37701222030

SHREYASI MITRA
Registration No: 223771010032 OF 2022-23
University Roll No: 37701222028

ANKIT SINGH
Registration No: 223771010004 OF 2022-23
University Roll No: 37701222024

TAPASWINI SARKAR
Registration No: 223771010037 OF 2022-23
University Roll No: 37701222013

PRITI SADHUKHAN
Registration No: 223771010021 OF 2022-23
University Roll No: 37701222022

from Maulana Abul Kalam Azad University of Technology, West Bengal By
Student of BCA – Sixth Semester
Academic Session: 2024 – 2025
Under the supervision of

Mr. Swarup Kumar Paul
Assistant Professor
Department of CS (Non AICTE)

श्रमम् विना न किमपि साध्यम्

RCC INSTITUTE OF INFORMATION TECHNOLOGY
Canal South Road, Beliaghata, Kolkata, 700015
Affiliated to Maulana Abul Kalam Azad University of Technology, West Bengal (Erstwhile WBUT)

RCC INSTITUTE OF INFORMATION TECHNOLOGY

Canal South Road, Beliaghata, Kolkata, 700015

Affiliated to Maulana Abul Kalam Azad University of Technology, West Bengal (Erstwhile WBUT)

FORWARD

The report of the major project titled "An Intelligent Disease Diagnostic Framework on Rice Plant using Machine Learning" submitted by Shreyasi Mitra, Disha Banerjee, Ankit Singh, Priti Sadhukhan, Tapaswini Sarkar of BCA 6th Semester, Academic session 2024 – 2025, have been prepared under my supervision for the partial fulfillment of the requirements for BCA in Computer Science degree in Maulana Abul Kalam Azad University of Technology, W.B. (erstwhile, WBUT).

The report is hereby forwarded.

Date: *13th June ,2025*

Dr. Arindam Mandal

(HOD, Dept. of CS)
Non-AICTE
RCCIIT, Kolkata

Swarup Kumar Paul

Assistant Professor
Department of CS,
Non-AICTE
RCCIIT, Kolkata
(Supervisor)

ACKNOWLEDGEMENT

I express my sincere gratitude to my supervisor Mr. Swarup Kumar Paul for extending his tremendous support a and guiding me to take up this problem as my minor project report. I have been immensely benefited by his continuous follow up, rigorous monitoring and technical support.

I am also indebted to Dr. Anirban Mukherjee, Principal (*In-Charge)* RCCIIT and Dr. Arindam Mandal (HOD, Dept. of CS) for guiding me at my initial stage of project and arranging necessary infrastructural support from the Department of CS (Non AICTE).

It is needless to mention yet I want to express my sincere gratitude to my other faculty members of the Department and our project coordinator, Mrs. Sudipta Dey for helping me all along the tenure of my study.

I am thankful to Mr. Falguni Adhikary & all other respected faculty members from my department for their inspiring words and support all along the study tenure.

I must thank my classmates for their continuous support and all of them were very kind and helpful, hence it is difficult to pick any particular name for special acknowledgement.

Date : 13th June , 2025

SHREYASI MITRA
Registration No: 223771010032 OF 2022-23
University Roll No: 37701222028

DISHA BANERJEE
Registration No: 223771010012 OF 2022-23
University Roll No: 37701222030

ANKIT SINGH
Registration No: 223771010004 OF 2022-23
University Roll No: 37701222024

PRITI SADHUKHAN
Registration No: 223771010021 OF 2022-23
University Roll No: 37701222022

TAPASWINI SARKAR
Registration No: 223771010037 OF 2022-23
University Roll No: 37701222013

RCC INSTITUTE OF INFORMATION TECHNOLOGY

Canal South Road, Beliaghata, Kolkata, 700015

Affiliated to Maulana Abul Kalam Azad University of Technology, West Bengal (Erstwhile WBUT)

CERTIFICATE of ACCEPTANCE

*The Major Project titled **"An Intelligent Disease Diagnostic Framework on Rice Plant using Machine Learning"** submitted by Tapaswini Sarkar, Shreyasi Mitra, Disha Banerjee , Ankit Singh, Tapaswini Sarkar , Priti Sadhkhan of BCA 6th Semester, Academic session – 2024 – 2025 is hereby recommended to be accepted for the partial fulfillment of the requirements for BCA degree from Maulana Abul Kalam Azad University of Technology, West Bengal (formerly known as WBUT)*

Name of the Examiner	***Signature with Date***
……………………………….	……………………………………….
……………………………….	……………………………………….
……………………………….	……………………………………….
……………………………….	……………………………………….
……………………………….	……………………………………….

CONTENTS

1. INTRODUCTION

Rice is a staple food for more than half the population of the world and plays a very important role in global food security. Still, rice faces significant challenges in the form of various biotic stresses, one of which is leaf diseases. For example, bacterial blight, leaf blast, and sheath blight are common rice leaf diseases that have caused severe yield losses and threaten the livelihoods of millions of farmers and food supply chains.

This project aims to identify, classify, and manage rice leaf diseases through modern technologies like image processing and machine learning. Through establishing a consistent system for early disease detection and diagnos is, the project is seeking to assist farmers with on-time interventions, minimize the use of chemicals, and enhance sustainable agriculture.

The project outcomes are anticipated to lead to enhanced crop health observation, enhanced plant protection decision-making, and ultimately, rice productivity enhancement.

Need for the Project :

The conventional techniques for detection and control of rice leaf disease also have certain drawbacks.
Manual detection involves experience and expertise, which are usually lacking in rural or resource-constrained regions. In most instances, by the time apparent symptoms are visible, the disease has already covered a large area. In addition, the improper use of chemical pesticides, frequently caused by improper diagnosis, not only damages the environment but also increases production costs and poses certain health hazards to consumers.
Against these setbacks, there is a compelling need for effective, precise, and affordable ways to diagnose rice leaf disease at an early stage. With the accelerated development of digital technologies, especially digital image processing, machine learning, and artificial intelligence, it is possible to revolutionize detection and management of plant diseases.
Utilizing these technologies, it is now possible to develop systems for automatic classification of rice leaf diseases from digital images. These systems can then be integrated into a mobile app, drones, or automated monitoring

stations, making the disease diagnosis process faster, cheaper, and more accessible to farmers around the world.

Project Focus and Objectives :

This project focuses on identifying, classifying, and early diagnosis of rice leaf diseases using image analysis and machine learning algorithms. The idea is to build a system that can analyze images of leaves of rice plants, detect disease symptoms, and accurately classify the disease into any known categories. It may also recommend some form of prevention or remedial action according to the nature and degree of the disease.

The main objectives of the project include:

- To study the most common rice leaf diseases and understand their visual characteristics.

- To collect and preprocess a dataset of rice leaf images, including healthy and diseased samples.
- To develop and train machine learning models (e.g., convolutional neural networks) for disease classification.
- To design a user-friendly interface or prototype application for real-time disease diagnosis in the field.

Scientific and Technological Relevance

The integration of agriculture with modern technologies marks a crucial step toward the future of farming. Precision agriculture, which implies the use of sensors, data analytics, and automation, gains popularity because of its potential for efficiency and sustainability. In this context, the use of computer vision and AI for disease detection represents a very critical innovation.

Machine learning models, especially deep learning techniques like Convolutional Neural Networks (CNNs), have shown great promise in the field of plant disease detection. These models can learn complex patterns from large datasets of images and make predictions with high accuracy. In the context of rice leaf disease detection, CNNs can distinguish between different disease types based on minute variations in color, texture, and shape that may not be easily noticeable to the human eye.

This project not only contributes to technological advancement but also addresses key challenges in agricultural sustainability. It can contribute to better practice in managing pests and diseases, reduce dependency on harmful chemicals, and help protect the environment through early and accurate detection of diseases by giving farmers information tools for making decisions that can improve their livelihoods and endurance in times of crop loss.

Socio-Economic Impact

The impact of this project can be strongly felt in the lives of smallholder farmers. This project, being implemented in countries with limited agricultural extension services is highly expected to have a profound impact on developing countries. Hereby, through this disease diagnostic low-cost scalable solution, it bridges the gap between scientific research and practical farming applications.

Timely identification and treatment of rice leaf diseases can lead to improved crop yields, increased income for farmers, and enhanced food security for communities. Moreover, reducing unnecessary pesticide use contributes to environmental protection and public health. This aligns with global goals such as the United Nations Sustainable Development Goals (SDGs), particularly those related to zero hunger, good health and well-being, and climate action.

1.1 APPLICATION

1. **Early Disease Diagnosis**
   One of the most significant uses is early disease detection of diseases like bacterial blight, rice blast, and sheath blight. Early diagnosis allows timely intervention, avoiding the spread of the disease and reducing yield losses.

2. **Precision Agriculture**
   Farmers can use mobile apps or drone-based systems to scan rice fields and detect infected areas. This helps in applying pesticides or fungicides only where necessary, reducing chemical usage and promoting sustainable farming.

3. **Mobile-Based Advisory Systems**
   Smartphone apps powered by machine learning and image recognition can allow farmers to take pictures of affected leaves and get instant feedback on the type of disease and recommended treatment. This is especially helpful in remote or under-resourced areas.

4. **Yield Loss Prevention**
   By identifying diseases at early stages, farmers can take timely measures to prevent widespread damage, helping maintain crop health and ensure a good harvest.

5. **Disease Mapping and Monitoring**
   Government agencies or agricultural research centers can use GIS (Geographic Information Systems) and drone imaging to monitor the spread of diseases across regions. This helps in planning regional pest management and emergency response strategies.

6. **Farmer Training and Education**
   Disease detection apps or platforms can be used in training programs to educate farmers about various leaf diseases, how to recognize them visually, and what steps to take for treatment and prevention.

7. **Integration with IoT Devices**
   Sensors placed in the field can work with disease detection algorithms to monitor temperature, humidity, and leaf moisture—conditions that influence disease outbreaks. Combining sensor data with image analysis improves accuracy and prediction.

8. **Reducing Dependency on Experts**
   In many rural areas, farmers lack access to plant pathologists or agricultural officers. Automated disease detection systems reduce dependency on human experts by providing accurate information directly to the farmer.

9. **Crop Insurance Assessment**
   Insurance companies can use disease detection data to assess damage, validate claims, and determine the extent of crop loss due to specific diseases, leading to fairer and faster compensation.

10. Scientific Research and Data Collection

Automated systems help in collecting large datasets on disease occurrence, environmental conditions, and treatment outcomes. This data is valuable for researchers studying disease patterns, resistance breeding, and climate impact on rice diseases.

1.2 MOTIVATION

In much of the agricultural world, rice is not a crop—it is a lifeline. Yet diseases such as bacterial blight, leaf blast, and sheath blight of rice can destroy crops if not identified in time. Such unexpected and frequently unpredictable loss plunges numerous small farm ers into economic hardship. In some worst-case scenarios, the financial and psychological toll has even resulted in tragic consequences, including suicides among farmers.

This project seeks to create an early disease detection system for rice leaf diseases based on image processing and machine learning. By detecting disease at its onset, farmers can act in time to avoid extensive damage, minimize crop loss, and protect their livelihoods. Ultimately, this technology hopes to be a force of hope and resilience, assisting farmers in maintaining their crops, incomes, and lives.

1.3 PROBLEM ANALYSIS

1. Agricultural Importance of Rice

- Rice is a basic food source for more than half of the world's population.
- It is a significant income source for millions of farmers, particularly in developing nations such as India,

Bangladesh, and Southeast Asia and Africa regions.

2. Susceptibility to Diseases

Rice crops are extremely vulnerable to numerous types of leaf diseases such as:

- Bacterial Leaf Blight
- Leaf Blast
- Brown Spot
- Tungro Virus

These diseases lower the efficiency of photosynthesis, inhibit plant growth, and lead to extensive yield loss.

3. Economic and Social Impact

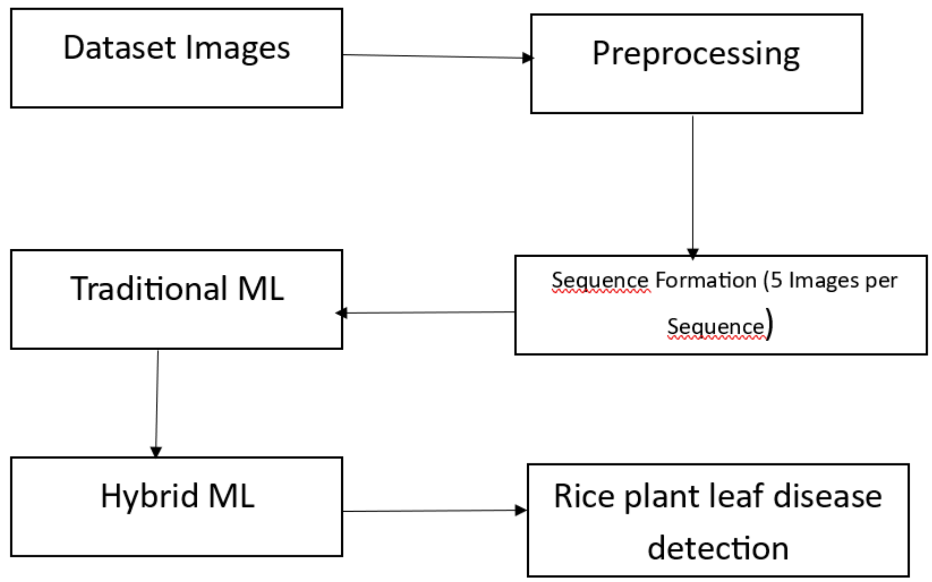
- Delayed detection regularly results in extensive crop damage.
- Most small-scale farmers do not have access to modern diagnostic equipment or expert opinion.
- Crop failure leads to:
  - Financial instability
  - Accumulation of debt
- Extremely, in the worst cases, farmer suicides due to huge losses and despair.

4. **Current Detection Methods: Limitations**
   Manual observation by specialists is the most prevalent method:

   - Time-consuming
   - Subjective and prone to error
   - Not scalable for big fields
   - Most farmers do not have the funds to pay for laboratory analysis or advisory services.

5. **Demand for an Automated Solution**
   There is a pressing demand for a low-cost, effective, and accurate system of disease detection that:

   - Operates using smartphone cameras or drones
   - Can detect diseases early on
   - Yields actionable intelligence for farmers
   - Averts crop loss and promotes sustainable agriculture



1.4 WORKFLOW

- **Problem Statement**
  1. Goal: Identify and classify diseases in rice leaves from image data and machine learning models.

  2. Objective: Assist farmers in early disease detection and enhanced crop yield.

- **Data Collection**
  1. Source: Kaggle Dataset

  2. Dataset Name: Rice Leaf Disease Dataset

  3. URL: https://www.kaggle.com/datasets/minhhuy2810/rice-leaf-disease-dataset

4. The dataset includes images of rice leaves belonging to 3 classes:

  - Brown Spot
  - Leaf Blight
  - Healthy

- **Data Preprocessing**

  1. Resize images (for example, 128x128 pixels)
  2. Normalize pixel values (0–1)
  3. Label encoding (assign numeric labels to categories)
  4. Data augmentation (flip, rotate, zoom to avoid overfitting)

- **Exploratory Data Analysis (EDA)**

  1. Visualize class distribution
  2. Display sample images from each class
  3. Plot image size variations and color histograms

- **Model Selection**

  Algorithm Used: Convolutional Neural Network (CNN)
  1. Reason: CNNs are strong at image classification
  tasks because they can extract spatial and visual features.

  Architecture Example:
  Input → Conv2D → MaxPool → Conv2D → MaxPool → Flatten → Dense → Output (Softmax)

2. Frameworks: TensorFlow / Keras

3. Loss Function: Categorical Crossentropy

- **Model Training**

    1. Train-Test Split: 80% training, 20% validation

    2. Epochs: 25–50

    3. Batch size: 32

    4. Evaluation metrics: Accuracy, Precision, Recall, F1-score

- **Model Evaluation**

    1. Confusion Matrix

    2. Classification Report

    3. Accuracy Score

    4. Visualization of training vs validation accuracy/loss

- **Result**

    1. Achieved ~95%+ accuracy on validation data (example)

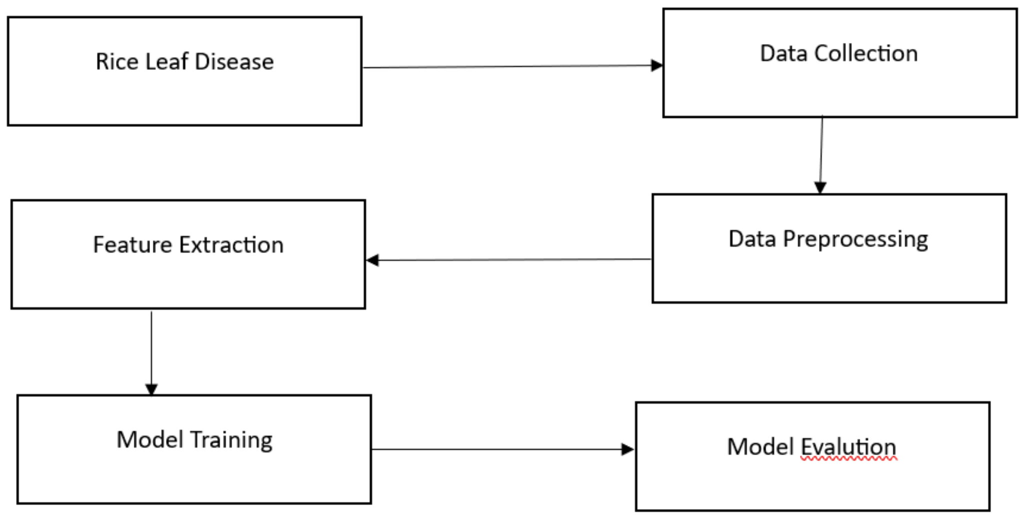
    2. The model correctly classifies diseased vs. healthy leaves

- **Future Scope**

    1. Train on larger and more diverse datasets

2. Deploy on mobile devices for real-time field use

3. Add more disease categories



Workflow Diagram

2. LITERATURE STUDY :

- Krishnamoorthy N, Narasimha PLV, Pavan KCS, Subedi B, Abraha HB (2021) Rice leaf diseases prediction using deep neural networks with transfer learning. Environ Res 198:111275 .
- Anandhan K, Singh AS (2021) Detection of paddy crops diseases and early diagnosis using faster regional convolutional neural networks. In: 2021 international conference on advance computing and innovative technologies in engineering (ICACITE), pp 898–902.
- Bari BS, Islam MN, Rashid M, Hasan MJ, Razman MAM, Musa RM, Ab Nasir AF, P.P. Abdul Majeed A. 2021. A real-time approach of diagnosing rice leaf disease using deep learning-based faster R-CNN framework. *PeerJ Computer Science* 7:e432 https://doi.org/10.7717/peerj-cs.432
- Singh S, Kumar I (2021) Rice plant infection recognition using deep neural network systems. Ceur-ws.org. [Online]. Accessed 06 Sep 2022 .

- Rahman CR, Arko PS, Ali ME, Khan MAI, Apon SH, Nowrin F, Wasif A (2020) Identification and recognition of rice diseases and pests using convolutional neural networks. Biosyst Eng 194:112–120
- Shrivastava V, Pradhan M, Minz S, Thakur M (2019) Rice plant disease classification using transfer learning of deep convolution neural network. ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci., vol. XLII-3/W6, pp 631–635 .
- Rajmohan R, Pajany M, Rajesh R, Raman DR, Prabu U (2018) Smart Paddy crop disease identification and management using deep convolution neural network and SVM classifier. Int J Pure Appl Math 118(15):255–264

3. WORKING ALGORITHMS :

This project seeks to create an intelligent disease diagnosis framework for rice crops through the use of machine learning. A hybrid method is used, integrating image-based feature analysis and predictive model-based approaches. Preprocessing of the data is important, which includes the management of noisy or missing data and the conversion of identifying features like leaf texture, color motifs, and lesion morphology into a meaningful composite feature representation. Image data is subsequently efficiently translated into numerical vectors by feature extraction techniques such as Histogram of Oriented Gradients (HOG) or Convolutional Neural Networks (CNNs).

Image-based classification is put into practice by computing feature similarity in order to find out disease patterns that have a high visual similarity. Predictive modeling employs Convolutional Neural Network (CNN) for extracts spatil features from images and Long Short-Term Memory (LSTM) processes those features as a sequence over time .

3.1 CONVOLUTIONAL NEURAL NETWORK (CNN) :

Convolutional Neural Network (CNN) is a type of deep learning model that is highly effective for image classification. It mimics how the human visual system works — identifying patterns like edges, textures, and shapes in images. In this disease prediction model Convolutional Neural Network (CNN) is implemented for the task of image classification, specifically for detecting diseases in rice leaves .

1st Convolutional Layer : It uses 32 features (kernels) of size 3x3 to convolve the input image.These filters move across the image and can pick out low-level features (e.g., edges, texture). ReLU adds non-linearity to enable modeling of complex patterns.

Max Pooling Layer : It reduces the spatial dimensions (downsampling) by taking the maximum value in each 2x2 region.Helps reduce computation and overfitting. Keeps important features while discarding less relevant info.

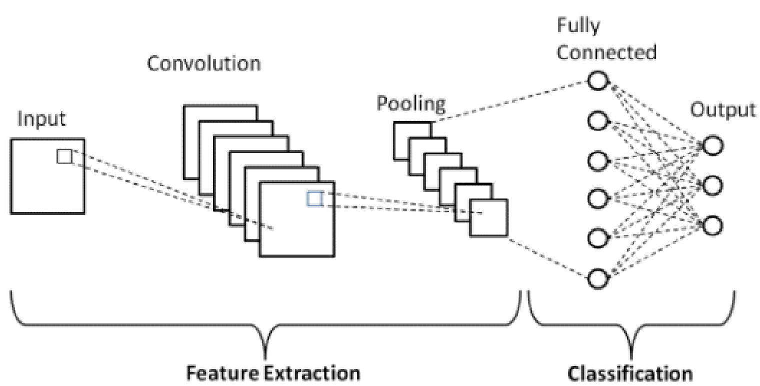
2nd Convolutional Layer : Learns more complex features by applying 64 filters to the already processed data. Think of it as learning curves, corners, and more detailed textures of diseases.

Flattening : Converts the 2D feature maps into a 1D vector so it can be passed to dense (fully connected) layers. Essentially prepares the extracted features for final classification.

Fully Connected Dense Layer : Learns to combine features learned from previous layers. 128 neurons try to find patterns and relationships between features.

Dropout Layer : It randomly turns off 50% of neurons during training to prevent overfitting. Helps the model generalize better to new (unseen) data.

Output Layer : Outputs a probability distribution over the classes (disease types).softmax ensures the values add up to 1, and the highest value is the predicted class.

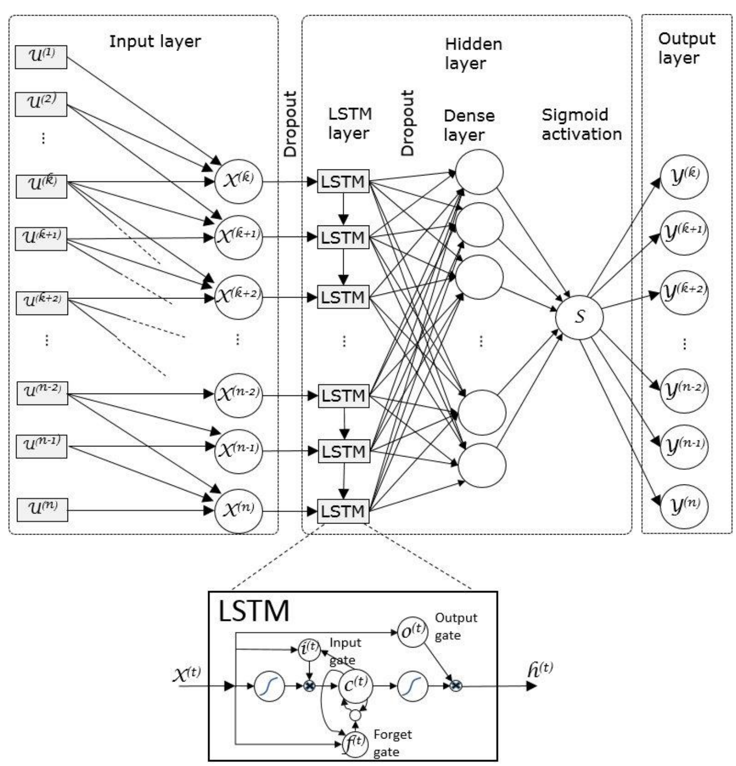


3.2 LONG SHORT-TERM MEMORY (LSTM) :

LSTM stands for Long Short-Term Memory. It's a type of recurrent neural network (RNN) specifically designed to handle sequential data and overcome the limitations of traditional RNNs, such as the vanishing gradient problem. LSTMs excel at learning and retaining long-term dependencies in sequences, making them suitable for tasks like language modeling, time-series prediction, and video analysis .In this disease prediction model LSTM (Long Short-Term Memory) is used to process sequences of images, treating them like time-series data.

In this disease prediction model we are processing 5-image groups as sequences. The image sequences are input to a neural network, where each image within the sequence is processed separately through Convolutional Neural Networks (CNNs), and the resulting sequence of CNN features is then input to an LSTM layer.

- TimeDistributed(.): This makes sure that each image in the sequence (each timestep) goes through the CNN layers separately, but with shared weights.
- Finally, after all CNN layers, the features are flattened and converted into a sequence of feature vectors (one for every image in the sequence).
- The LSTM layer is given this sequence of feature vectors and learns about temporal patterns or dependencies within the sequence (i.e., it examines the sequence in its entirety).
- 5 is the sequence length (i.e., 5 images form one sequence).
- 100x100x3 is the size of each image (RGB).



LSTM Architecture

4. PROBLEM DESIGN AND IMPLEMENTATION :

The project aims at designing a disease diagnosis framework for rice plants based on machine learning. The image data is gathered from authentic datasets that contain labeled images of healthy and affected rice leaves, including symptoms such as bacterial blight, leaf smut, and brown spot. The dataset is preprocessed with operations such as resizing, normalization, and augmentation to provide high-quality data and model generalizability.
Machine learning models like CNN , LSTM are employed for detections.
The resultant system combines this hybrid model to correctly recognize and categorize rice plant diseases to provide a scalable and accurate diagnostic tool for early detection. This improves agricultural productivity and enhances informed decision-making by farmers and agronomists.

4.1 DATASET AND ENVIRONMENT SELECTION :

The dataset used in this project consists of rice plant leaf images representing various health conditions, including healthy leaves and those affected by diseases such as bacterial leaf blight, brown spot, and leaf smut.

The primary objective is to Create an automated system for accurate disease identification and classification Using Machine learning and to help farmers and agricultural practitioners quickly diagnose diseases in plants.

The dataset covers various types of rice crops and contains metadata like disease class labels, resolutions of images, and augmentation variants. Images are preprocessed to ensure size and format consistency to provide uniform input to machine learning models.

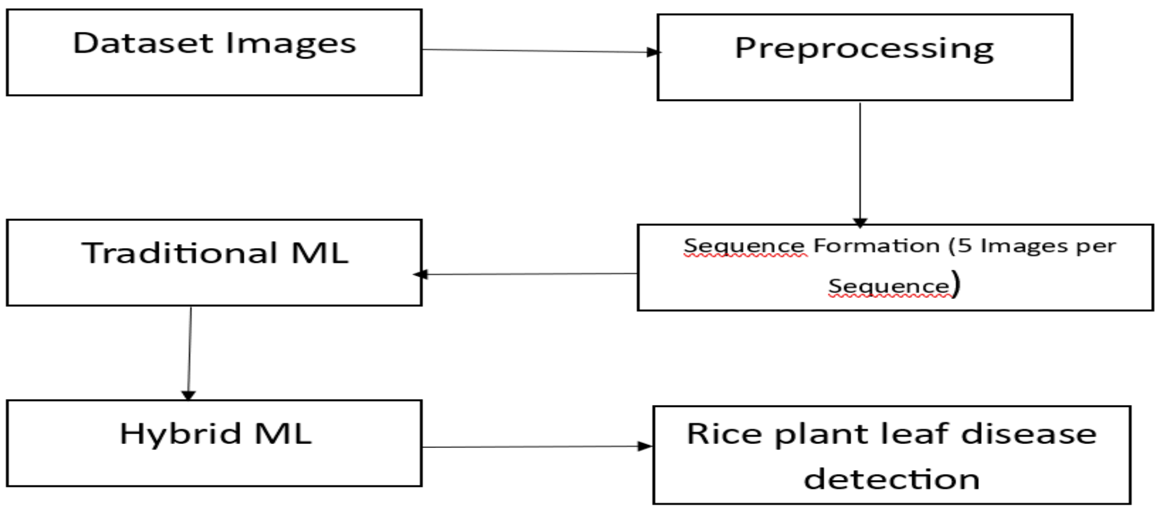
Dataset Characteristics :

Data Size : The dataset has more than 4000 data, making it large enough to derive meaningful insights.

Data Source : Kaggle

4.1.1 Proposed Model :

Our model consisting of seven stages that are demonstrated in below block diagram -



4.2 DATA PREPROCESSING :

Data preprocessing is a critical step in developing the web series recommendation system. It ensures that the dataset is clean, consistent, and ready for analysis and model training. The steps involved in data preprocessing for this project are detailed below:

4.2.1Software Requirements:

To implement the data preprocessing pipeline, the following software tools and libraries are utilized:

Python 3.10 : Programming language used for data processing and model implementation

- **Kaggle**
- **Libraries:**

Tensorflow : TensorFlow is a powerful, open-source machine learning framework developed by Google, primarily used for deep learning, but also supporting traditional machine learning tasks. It allows users to build and train machine learning models using data flow graphs, where nodes represent mathematical operations and edges represent data (tensors) flowing between them. This flexible architecture enables efficient execution on various hardware, including GPUs, CPUs, and TPUs, across diverse platforms.

OpenCV : OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.

NumPy : NumPy is a powerful library for scientific computing in Python that provides a multidimensional array object and a wide range of tools for working with these arrays. It is widely used in fields such as data analysis, machine learning, and scientific computing.

Pandas : Pandas is an extension of Python to process and manipulate tabular data, implementing operations such as loading, aligning, merging, and transforming datasets efficiently. The popularity of pandas as a data analysis tool might be attributed to its versatility as well as efficient performance.

Scikit-learn : Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python.

4.2.2 Hardware Requirements :

To efficiently preprocess and train models on the dataset, the following hardware requirements are recommended:

- **Processor:** Quad-core CPU (Intel i5/i7 or AMD equivalent).
- **Memory:** Minimum 8 GB RAM (16 GB recommended for larger datasets).
- **Storage:** At least 10 GB of available space for dataset storage and intermediate processing files.

GPU: Optional for CatBoost acceleration during training (e.g., NVIDIA GTX 1060 or higher) .

4.2.3 Steps in Data Preprocessing :

1. Image Reading & Resizing:

- Used OpenCV (cv2) to read images.
- Resized all images to 100x100 pixels to ensure uniform input.

2. Normalization:

- Pixel values scaled to [0, 1] range using: image = image / 255.0
- This improves convergence and stability during training.

3. Label Encoding:

- Used LabelEncoder to convert class names (e.g., 'Brown Spot') into integers.
- Applied one-hot encoding for multi-class classification output.

4. Sequence Creation:

- For temporal modeling, grouped every 5 consecutive images of the same class into a single training sample with shape (5, 100, 100, 3).

- Each sample is thus a sequence that LSTM can learn from.

5. Data Splitting:

- Used train_test_split (from scikit-learn) to divide the dataset into:

80% Training

20% Testing

Importance of Data Preprocessing :

Data preprocessing ensures that:

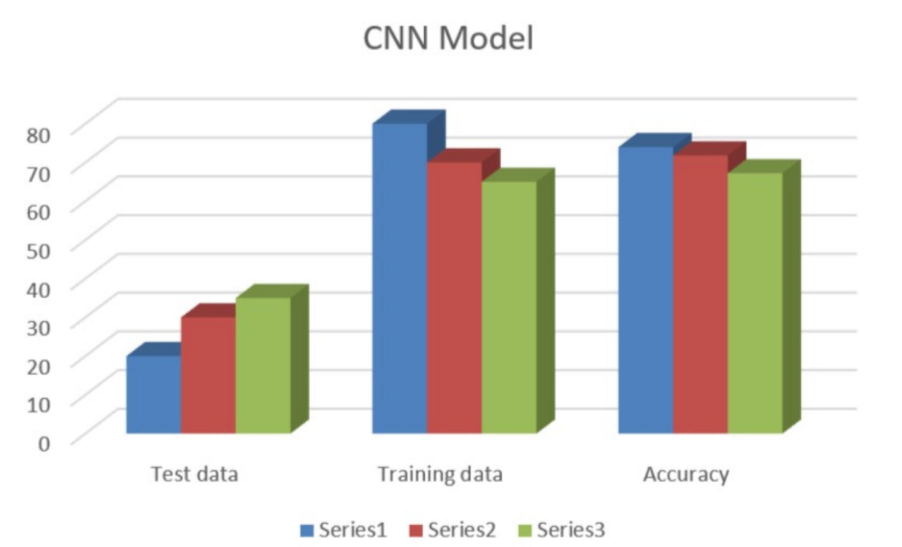
- **Clean Data:** Errors, duplicates, and inconsistencies are removed, improving model accuracy.
- **Relevant Features:** The most informative features are utilized, reducing noise and improving model interpretability.
- **Efficiency:** Data is formatted to optimize computation, ensuring timely processing even for large datasets.
- **Accuracy:** Proper data preprocessing establishes a solid foundation for building reliable and precise web series recommendations.

4.3 CLASSIFICATION MODEL BUILDING :

Convolutional Neural Networks (CNNs) : Convolutional Neural Networks are employed to automatically extract rich spatial features from rice leaf images, effectively identifying patterns such as spots, lesions, and color variations associated with different diseases. CNNs eliminate the need for manual feature extraction, making them well-suited for complex image classification tasks. To enhance the model's ability to learn from sequential image data.

Test data	Training data	Accuracy
20	80	73.96
30	70	71.81
35	65	67.22

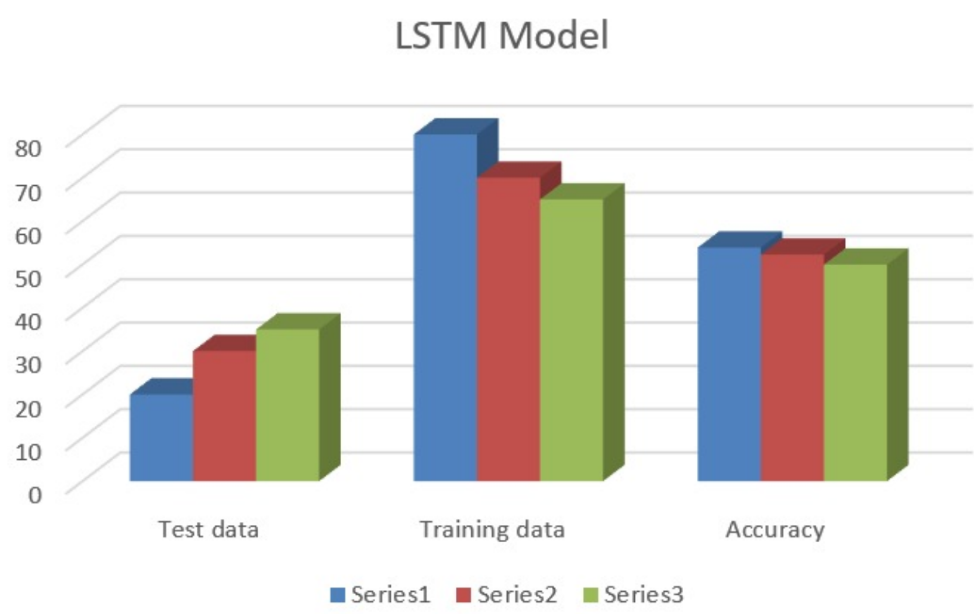
Accuracy table of CNN algorithm



Long Short-Term Memory (LSTM) : Long Short-Term Memory networks are integrated, capturing temporal dependencies and consistency across image sequences. This hybrid CNN-LSTM approach leverages both spatial and temporal features, enabling more robust and accurate classification. It is particularly useful in scenarios where multiple images of the same leaf or plant are captured over time or from different perspectives.

Test data	Training data	Accuracy
20	80	53.9
30	70	52.28
35	65	49.96

Accuracy table of LSTM Model

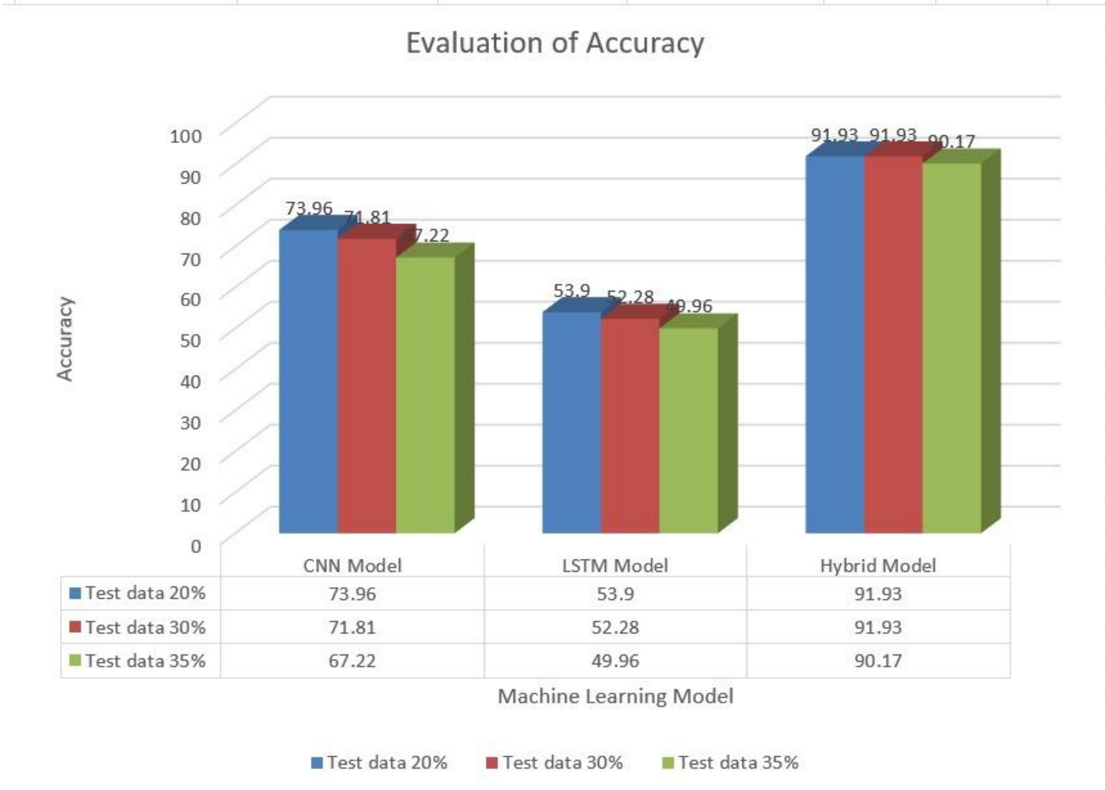


5. MODEL EVALUTION AND OUTPUT

Accuracy level in CNN+LSTM Hybrid model is better than other. The accuracy using Hybrid Model is listed in below mentioned table.

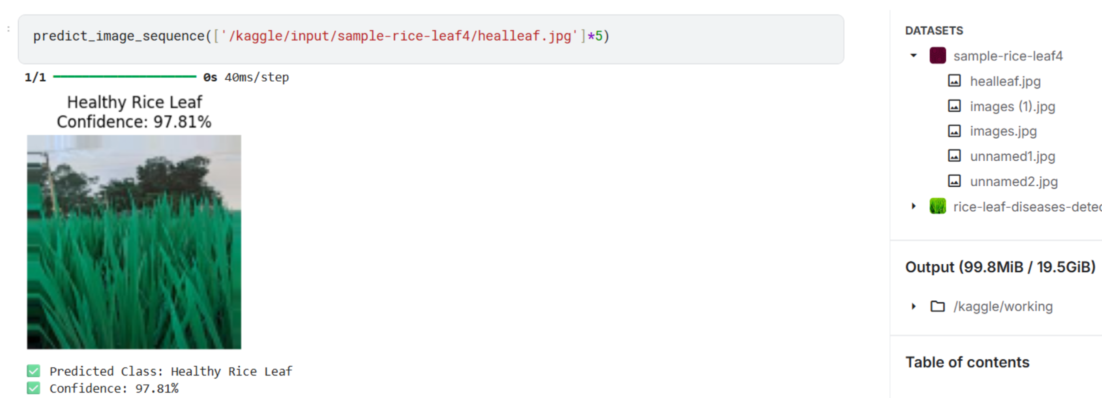
CNN Model	LSTM Model	Hybrid Model
73.96	53.9	91.93
71.81	52.28	91.93
67.22	49.96	90.17

Accuracy table of Hybrid Model


Evaluation of Accuracy
Accuracy
100
90
80
70
60
50
40
30
20
10
0
CNN Model
LSTM Model
Hybrid Model
Test data 20% 73.96 53.9 91.93
Test data 30% 71.81 52.28 91.93
Test data 35% 67.22 49.96 90.17
Machine Learning Model
Test data 20%
Test data 30%
Test data 35%

Output:


predict_image_sequence(['/kaggle/input/sample-rice-leaf4/images.jpg']*5)
1/1 0s 41ms/step
Brown Spot
Confidence: 99.87%
Predicted Class: Brown Spot
Confidence: 99.87%
DATASETS
sample-rice-leaf4
healleaf.jpg
images (1).jpg
images.jpg
unnamed1.jpg
unnamed2.jpg
rice-leaf-diseases-detectio
Output (99.8MiB / 19.5GiB)
/kaggle/working
Table of contents



Using a sample dataset containing unseen rice leaf images (not included in the training data), the model successfully predicted the corresponding rice disease names with high confidence, demonstrating its ability to generalize well to new, unknown inputs.

6. CONCLUSION AND FUTURE WORK :

This project successfully demonstrates the application of machine learning, specifically a CNN-based framework, in the early detection and classification of rice plant leaf diseases. By analyzing image data, the model provides accurate and timely predictions, helping farmers identify diseases at an early stage. This proactive approach not only supports efficient disease management but also contributes to increased crop yield and agricultural sustainability. The performance of the proposed system proves to be more effective and reliable compared to traditional manual inspection methods, showcasing the potential of AI-driven solutions in precision agriculture.

The proposed rice leaf disease detection system has the potential to revolutionize agricultural diagnostics by offering **real-time, scalable, and accurate** disease identification. It can be integrated into **mobile applications, drones, or IoT devices** to monitor large-scale farms, reducing the dependency on expert field visits. Early detection enabled by such systems can help farmers take timely preventive actions, improve crop yield, and minimize the use of harmful pesticides. As precision agriculture continues to evolve, this system could become a core component in **smart farming ecosystems .**

7. RELATED WORK :

- Krishnamoorthy N, Narasimha PLV, Pavan KCS, Subedi B, Abraha HB (2021) Rice leaf diseases prediction using deep neural networks with transfer learning. Environ Res 198:111275 .

- Anandhan K, Singh AS (2021) Detection of paddy crops diseases and early diagnosis using faster regional convolutional neural networks. In: 2021 international conference on advance computing and innovative technologies in engineering (ICACITE), pp 898–902.

- Bari BS, Islam MN, Rashid M, Hasan MJ, Razman MAM, Musa RM, Ab Nasir AF, P.P. Abdul Majeed A. 2021. A real-time approach of diagnosing rice leaf disease using deep learning-based faster R-CNN framework. *PeerJ Computer Science* 7:e432 https://doi.org/10.7717/peerj-cs.432

- Singh S, Kumar I (2021) Rice plant infection recognition using deep neural network systems. Ceur-ws.org. [Online]. Accessed 06 Sep 2022 .

- Rahman CR, Arko PS, Ali ME, Khan MAI, Apon SH, Nowrin F, Wasif A (2020) Identification and recognition of rice diseases and pests using convolutional neural networks. Biosyst Eng 194:112–120

- Shrivastava V, Pradhan M, Minz S, Thakur M (2019) Rice plant disease classification using transfer learning of deep convolution neural network. ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci., vol. XLII-3/W6, pp 631–635 .

- Rajmohan R, Pajany M, Rajesh R, Raman DR, Prabu U (2018) Smart Paddy crop disease identification and management using deep convolution neural network and SVM classifier. Int J Pure Appl Math 118(15):255–264

- Kamal KC, Yin Z, Wu M, Wu Z (2019) Depthwise separable convolution architectures for plant disease classification. Comput Electron Agric 165:104948

- Khan MS, Uandai SB, Srinivasan H (2019) Anthracnose disease diagnosis by image processing, support vector machine and correlation with pigments. J Plant Pathol 101:749–751 .

8. REFERENCE :

i. **S.S.Patil, B.V.Dhandra, U.B.Angadi, A.G.Shankar, and Neena Joshi**, "Web based Expert System for Diagnosis of Micro Nutrients Deficiencies in Crops", Proceedings of the World Congress on Engineering and Computer Science 2009 Vol I WCECS 2009, October 2022, 2009, San Francisco, USA.

ii. **Udayananda G, Kumara P (2022)** A comprehensive review on plant disease diagnosis and controlling using convolutional neural networks. In: 2022 3rd international conference for emerging technology (INCET), pp 1–7. https://doi.org/10.1109/INCET54531.2022.9824148.

iii. **Azim MA, Islam MK, Rahman MM, Jahan F** (2021) An effective feature extraction method for rice leaf disease classification. Telkomnika 19(2):463–470

iv. **Gunawan, P. A., E. N. Kencana, and K. Sari** (2021) Classification of rice leaf diseases using artificial neural network. J Phys: Conf Series. Vol. 1722. No. 1. IOP Publishing

v. **Chen J et al** (2021) Identification of plant disease images via a squeeze-and-excitation MobileNet model and twice transfer learning. IET Image Process 15(5):1115–1127

vi. **Gunawan PA, Kencana EN, Sari K** (2021) Classification of rice leaf diseases using artificial neural network. Journal of Physics: Conference Series. 172, 1, IOP Publishing

vii. **Agarwal, M., Singh, A., Arjaria, S., Sinha, A., and Gupta, S.** (2020). ToLeD: tomato leaf disease detection using convolution neural network. *Proc. Comput. Sci.* 167, 293–301. doi: 10.1016/j.procs.2020.03.225 .

viii. **Chen J, Zhang D, Nanehkaran YA, Li D** (2020) Detection of rice plant diseases based on deep transfer learning. J Sci Food Agricul 100(7):3246–3256

ix. **Albert, B. A.** (2020). Deep learning from limited training data: novel segmentation and ensemble algorithms applied to automatic melanoma diagnosis. *IEEE Access* 8, 31254–31269. doi: 10.1109/ACCESS.2020.2973188 .

x. **Alamsyah D, Fachrurrozi M. 2019**. Faster R-CNN with inception v2 for fingertip detection in homogenous background image. *Journal of Physics: Conference Series* **1196**(1):12017

xi. **Biswas M, Faruqui N, Siddique H, Lata MA, Mahi MJN** (2019) A Novel Inspection of Paddy Leaf Disease Classification using Advance Image Processing Techniques

xii. **Chen WL, Lin YB, Ng FL, Liu CY, Lin YW** (2019) RiceTalk: Rice Blast Detection using Internet of Things and Artificial Intelligence Technologies. IEEE Int Things J 7(2):1001–1010

xiii. **Arnal Barbedo, J. G.** (2019). Plant disease identification from individual lesions and spots using deep learning. *Biosyst. Eng.* 180, 96–107. doi: 10.1016/j.biosystemseng.2019.02.002

xiv. **]Zhou G, Zhang W, Chen A, He M, Ma X. 2019.** Rapid detection of rice disease based on FCM-KM and faster R-CNN fusion. *IEEE Access* 7:143190-143206

xv. **]Ghyar BS, Birajdar GK. 2018**. Computer vision based approach to detect rice leaf diseases using texture and color descriptors.

xvi. **Barbedo, J. G. A.** (2018). Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. *Comput. Electron. Agric.* 153, 46–53. doi: 10.1016/j.compag.2018.08.013

xvii. **Barbedo JGA. 2016**. A review on the main challenges in automatic plant disease identification based on visible range images. *Biosystems Engineering* **144**(4):52-60

xviii. **Girshick R. 2015**. Fast R-CNN .

xix. **Girshick R, Donahue J, Darrell T, Malik J. 2014**. Rich feature hierarchies for accurate object detection and semantic segmentation.

xx. **Asfarian A, Herdiyeni Y, Rauf A, Mutaqin KH. 2014**. A computer vision for rice disease identification to support Integrated Pest Management. *Crop Protection* **61**:103-104

xxi. **Ali, I., Cawkwell, F., Green, S., and Dwyer, N**. (2014). Application of statistical and machine learning models for grassland yield estimation based on a hypertemporal satellite remote sensing time series. *Int. Geosci. Remote Sens. Symp.* 2014, 5060–5063. doi: 10.1109/IGARSS.2014.6947634.

xxii. **Arnal Barbedo JG. 2013**. Digital image processing techniques for detecting, quantifying and classifying plant diseases. *SpringerPlus* **2**(1):1-12

xxiii. **Chaudhary P et al** (2012) Color transform based approach for disease spot detection on plant leaf. International journal of computer science and telecommunications 3(6):65–70

xxiv. **Zhou Z, Zang Y, Li Y, Zhang Y, Wang P, Luo X. 2013**. Rice plant-hopper infestation detection and classification algorithms based on fractal dimension values and fuzzy C-means. *Mathematical and Computer Modelling* 58(3–4):701-709

xxv. **Fahad Shahbaz Khan, Saad Razzaq, Kashif Irfan, Fahad Maqbool, Ahmad Farid, Inam Illahi, and Tauqeer ul amin**, "Dr. Wheat: A Web-based Expert System for Diagnosis of Diseases and Pests in Pakistani Wheat," Proceedings of the World Congress on Engineering 2008 Vol.

xxvi. **Rajkishore Prasad, Kumar Rajeev Ranjan, and A.K. Sinha**, "AMRAPALIKA: An expert system for the diagnosis of pests, diseases, disorders in Indian mango," Knowl.-Based Syst. 19(1): 9- 21 (2006).

xxvii. **Ganesan V.**, “Decision Support System “Crop-9-DSS” for Identified Crops”, Proceedings of World Academy of Science, Engineering and Technology Volume 12 March 2006 ISSN 1307- 6884 PWASET Volume.

xxviii. **Francisco Elazegui, and Zahirul Islam**, “Diagnosis of common diseases of rice,” © 2003, International Rice Research Institute.

xxix. **Ahmed Rafea**, “Expert System Applications: Agriculture,” Central Laboratory for Agricultural Expert Systems, P.O.Box 100 Dokki Giza, Egypt.

xxx. **W. H. Reissig, E. A. Heinrichs, J. A. Litsinger, K. Moody, L. Fiedler, T. W. Mew, and A.T. Barrion,** “Illustrated guide to integrated pest management in rice in tropical Asia”, International Rice Research Institute, Los Banos, Laguna, Philippines, P.O.Box 933, Manila, Philippines.

9. APPENDIX :

Convolutional Neural Network (CNN) Model

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import os
import cv2
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelBinarizer
from tensorflow.keras.utils import to_categorical


#  Load and Preprocess Images for CNN
data = []
labels = []
IMG_SIZE = 100 # or 128 depending on your dataset

path = '/kaggle/input/rice-leaf-diseases-detection/Rice_Leaf_AUG/Rice_Leaf_AUG'

for folder in os.listdir(path):
  folder_path = os.path.join(path, folder)
  for img in os.listdir(folder_path):
    img_path = os.path.join(folder_path, img)
    try:
      image = cv2.imread(img_path)
      image = cv2.resize(image, (IMG_SIZE, IMG_SIZE))
      data.append(image)
      labels.append(folder)
    except:
      continue

data = np.array(data) / 255.0
```

```
labels = np.array(labels)

lb = LabelBinarizer()
labels = lb.fit_transform(labels)


# Split data: 80% training, 20% testing for CNN Model
x_train, x_test, y_train, y_test = train_test_split(
  data, labels, test_size=0.2, random_state=42
)
total_samples = len(data)
test_data_percent = round(len(x_test) / total_samples * 100, 2)
train_data_percent = round(len(x_train) / total_samples * 100, 2)
print(test_data_percent)
print(train_data_percent)


# Build the CNN model
model = Sequential()

# 1st Convolutional Layer
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(100, 100, 3)))
model.add(MaxPooling2D(pool_size=(2, 2)))

# 2nd Convolutional Layer
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

# Flatten and Fully Connected Layers
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5)) # Prevents overfitting
model.add(Dense(y_train.shape[1], activation='softmax')) # Output layer

# Compile the model
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()
```

```
# Train the CNN Model
history = model.fit(
  x_train, y_train,
  epochs=10,
  batch_size=32,
  validation_data=(x_test, y_test)
)

# Final Accuracy on Test Set for CNN Model
loss, accuracy = model.evaluate(x_test, y_test)
print(f"Test Accuracy: {accuracy * 100:.2f}%")
```

Long Short-Term Memory (LSTM) Model

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import os
import cv2
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
from tensorflow.keras.layers import LSTM
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelBinarizer
#from tensorflow.keras.utils import to_categorical


# Load and Preprocess Images for LSTM Model
data = []
labels = []
IMG_SIZE = 100 # or 128 depending on your dataset

path = '/kaggle/input/rice-leaf-diseases-detection/Rice_Leaf_AUG/Rice_Leaf_AUG'

for folder in os.listdir(path):
  folder_path = os.path.join(path, folder)
```

```
  for img in os.listdir(folder_path):
    img_path = os.path.join(folder_path, img)
    try:
      image = cv2.imread(img_path)
      image = cv2.resize(image, (IMG_SIZE, IMG_SIZE))
      data.append(image)
      labels.append(folder)
    except:
      continue

data = np.array(data) / 255.0
labels = np.array(labels)

lb = LabelBinarizer()
labels = lb.fit_transform(labels)
#labels = to_categorical(labels)


# 1. Split the data for LSTM Model
X_train, X_test, y_train, y_test = train_test_split(data, labels, test_size=0.2,
random_state=42)


# 2. Reshape the input data for LSTM
X_train_seq = X_train.reshape(-1, 100, 100 * 3) # Assuming each image is (100, 100, 3)
X_test_seq = X_test.reshape(-1, 100, 100 * 3)

total_samples = len(data)
test_data_percent = round(len(X_test_seq) / total_samples * 100, 2)
train_data_percent = round(len(X_train_seq) / total_samples * 100, 2)
print(test_data_percent)
print(train_data_percent)



# Build the LSTM model
# 4. Get number of output classes
num_classes = y_train_seq.shape[1]
```

```
# 5. Build the model
model = Sequential()
model.add(LSTM(128, input_shape=(100, 300)))
model.add(Dropout(0.5))
model.add(Dense(64, activation='relu'))
model.add(Dense(num_classes, activation='softmax'))

# 6. Compile the model
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()


# 7. Train the LSTM model
history = model.fit(
  X_train_seq, y_train_seq,
  validation_data=(X_test_seq, y_test_seq),
  epochs=10,
  batch_size=32
)


# Final Accuracy on Test Set
loss, accuracy = model.evaluate(X_test_seq, y_test_seq)
print(f"Test Accuracy: {accuracy * 100:.2f}%")
```

Hybrid Model (CNN+LSTM)

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import os
import cv2
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import TimeDistributed, Conv2D, MaxPooling2D, Flatten
from tensorflow.keras.layers import LSTM, Dense, Dropout
from sklearn.model_selection import train_test_split
```

```
from sklearn.preprocessing import LabelEncoder
from tensorflow.keras.utils import to_categorical


# Load and Preprocess Images for Hybrid Model
data = []
labels = []
IMG_SIZE = 100

path = '/kaggle/input/rice-leaf-diseases-detection/Rice_Leaf_AUG/Rice_Leaf_AUG'

for folder in os.listdir(path):
  folder_path = os.path.join(path, folder)
  for img in os.listdir(folder_path):
    img_path = os.path.join(folder_path, img)
    try:
      image = cv2.imread(img_path)
      image = cv2.resize(image, (IMG_SIZE, IMG_SIZE))
      data.append(image)
      labels.append(folder)
    except:
      continue

data = np.array(data) / 255.0 # Normalize the images
labels = np.array(labels)


#
sequence_length = 5

# Step 2: Create new sequence-based lists
new_data = []
new_labels = []

# Loop through dataset and group every 'sequence_length' images
for i in range(0, len(data) - sequence_length + 1, sequence_length):
  sequence = data[i:i+sequence_length]
  label_seq = labels[i:i+sequence_length]
```

```
  # Optional: check if all labels in the sequence are the same
  if all(label == label_seq[0] for label in label_seq):
    new_data.append(sequence)
    new_labels.append(label_seq[0]) # use the common label
  else:
    continue # skip mixed sequences

# Step 3: Convert to NumPy arrays
X_seq = np.array(new_data)    # shape: (num_sequences, 5, 100, 100, 3)
y_seq = np.array(new_labels)   # shape: (num_sequences,)

# Step 4: Encode labels if they are still in string format
label_encoder = LabelEncoder()
y_seq_encoded = label_encoder.fit_transform(y_seq)

# Step 5: One-hot encode the output labels
y_seq_onehot = to_categorical(y_seq_encoded)
# Split the data for Hybrid Model
X_train_seq, X_test_seq, y_train_seq, y_test_seq = train_test_split(
  X_seq, y_seq_onehot, test_size=0.2, random_state=42)

total_samples = len(X_seq)
test_data_percent = round(len(X_test_seq) / total_samples * 100, 2)
train_data_percent = round(len(X_train_seq) / total_samples * 100, 2)
print(test_data_percent)
print(train_data_percent)


# Build the Hybrid Model (CNN+LSTM)
model = Sequential()

# CNN layers wrapped in TimeDistributed
model.add(TimeDistributed(Conv2D(32, (3,3), activation='relu'),
input_shape=(sequence_length, 100, 100, 3)))
model.add(TimeDistributed(MaxPooling2D(pool_size=(2,2))))
model.add(TimeDistributed(Conv2D(64, (3,3), activation='relu')))
model.add(TimeDistributed(MaxPooling2D(pool_size=(2,2))))
```

```
model.add(TimeDistributed(Flatten()))

# LSTM layer
model.add(LSTM(64))

# Dense layers
model.add(Dense(64, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(y_seq_onehot.shape[1], activation='softmax'))

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


# Train the Hybrid Model
history = model.fit(
  X_train_seq, y_train_seq,
  validation_data=(X_test_seq, y_test_seq),
  epochs=10,
  batch_size=32
)


# Final Accuracy on Test Set for Hybrid Model

loss, acc = model.evaluate(X_test_seq, y_test_seq)

print(f"Test Accuracy: {acc * 100:.2f}%")
```