TRIBHUVAN UNIVERSITY

FACULTY OF SCIENCE AND TECHNOLOGY

A Project Proposal

On

"Rice Leaf Disease Detection and Classification"

Submitted to

Department of Statistics and Computer Science

Patan Multiple Campus

In partial fulfillment of the requriments for Bachelor Degree in Computer science and Information Technology

Submitted By:

Saroj Pandey, 23084/076

Bimal Subedi, 23013/076

Remanta Paudel Khatri, 23062/076

Supervisor:

Nawaraj Poudel

Assistant Professor

ABSTRACT

Global farming communities still face a serious threat from crop losses brought on by insect pests and diseases, despite advancements in sustainable development and the growing use of smartphones, drones, satellites, and other information communication technologies for data collection and analysis in decision-making. When it comes to rice farming, diseases and insect pest infestations can account for up to 37% of economic losses. Not only can prompt and accurate diagnosis and treatment of these problems reduce crop losses, but they can also support sustainable agriculture.

The use of information communication technologies for information sharing has increased as a result of the growing availability of mobile internet services in rural areas. Smallholder farmers in economically disadvantaged areas are using ICT-based tools more frequently to diagnose diseases and insect pests, which helps them make well-informed decisions for effective pest management. Estimating the severity, impact, or incidence of a disease requires early classification and detection; this preventive measure not only minimizes crop production losses but also lessens the need for pesticides, which boosts the economy as a whole.

This project aims to develop a user-friendly web-based application that will help to identify the common diseases persisting in rice cultivation. It allows users to upload images of rice leaves via a simple interface and get insights into the disease affecting the plant. Behind the scenes, a deep learning model was built using a Convolutional Neural Network and trained on around 5000 images belonging to 8 disease classes categorized as *Narrow Brown Spot, Bacterial Leaf Blight, Healthy, Leaf Scaled, Leaf Blast, Tungro, Brown Spot, and Sheath Blight* is employed which provides response via Application Programming Interface (API) based on the input image provided by the user.

Keywords:

Rice Disease Classification and Detection, Convolutional Neural Network, Early Detection, Economic Impact, Precision Agriculture

Table of Contents

List of figures

CHAPTER 1

INTRODUCTION

1.1 Introduction

Nepal is a land of agriculture and rice (scientifically known as *Oryza sativa*) is a major crop that is crucial to the nation's agriculture and food security. Nepal offers a suitable environment for rice cultivation due to its varied agro-climatic conditions and varying altitudes. Rice is grown throughout Nepal and can be found all over Nepal, particularly in the hill districts and Terai regions. About 73% of rice is produced in the Terai, 24% in the hills, and 4% in the high hills. Rice is critical to food and nutrition security (67% of total cereal consumption and 23% of protein intake), employment, and income for farmers in addition to its contribution to the economy; e.g. 20% to Agricultural Gross Domestic Product and 7% to GDP [1]. Due to its fertile soil and hospitable climate, Terai produces the majority of the rice. For the Nepalese people, rice is a significant source of nutrition and helps them meet their dietary needs and consumed in various forms like steamed rice, rice pudding and various rice-based snacks.

Rice farming goes beyond a mere agricultural practice in many countries, like Nepal, where it is deeply interlinked with the culture and livelihoods of the people, serving as a fundamental means of survival. However, a number of illnesses that can completely wipe out crops if not properly diagnosed and treated put the health and yield of rice plants in danger. The impact of rice leaf diseases can be particularly severe in Nepal, where agriculture is the foundation of the economy, hurting not only farmer incomes but also the country's capacity to feed itself. The common rice leaf diseases found are Bacterial Leaf Blight, Brown Spot, Leaf Blast, Leaf Streak, Leaf Scald, Narrow Brown Spot, and Sheath Blight.

With the rapid advancement of technology and the growing prominence of machine learning, there is a unique opportunity to leverage artificial intelligence to effectively identify these diseases and address a serious global concern. This technological intervention has the potential to completely change the way rice is grown, improving crop management and protecting Nepali farmers' livelihoods.

1.1.1 Motivation

These days, tablets and smartphones are so commonplace that almost everyone owns one of these gadgets. Web-based applications are now a necessary part of our everyday lives in today's digital environment, and users access them from these gadgets. Taking full advantage of the broad availability of web connectivity and robust computing power, this project presents an efficient approach and a workable system for web-based application for detecting the common rice leaf diseases. Through the use of machine learning techniques, this project provides a more insightful, and user-friendly approach via the web interface in addition to improving the disease detection process. By providing farmers with timely and accurate information, this project hopes to enable proactive measures to protect rice crops.

1.2 Problem statement

In Nepal, rice leaf diseases pose a significant threat to rice farmers' livelihoods by causing huge yield losses and negatively impacting the supply of this essential staple crop. The current approach to identifying and managing rice leaf diseases is based on manual and subjective assessments, which often lead to delayed responses and sub-optimal outcomes. Furthermore, the wide variety of diseases and the requirement for timely intervention necessitate an effective and precise disease classification system. This proposal aims to provide a solution to this pressing issue by utilizing machine learning to identify rice leaf diseases quickly and accurately, thus empowering farmers and researchers to protect their crops.

1.3 Objectives

The main objective of this project *KrishiDhaaan* is:

- To develop a web application for uploading and processing images of rice leaves.
- To create and train a machine learning model for classifying rice leaf diseases.
- To provide users with accurate disease classification results and promote early detection.

1.4 Scopes and Limitations

The primary scope of the application is to provide a thorough method for identifying different rice leaf diseases. The application is made to be flexible enough to accommodate different types of rice farming in different parts of the world. A key focus is on providing farmers and other agricultural stakeholders a user-friendly web interface. By giving farmers timely and useful information, the system aims to enable them to respond proactively to diseases that are identified. However, some of the limitations are as follows:

1. The quality of the input data affects how accurately diseases can be detected. So, there might be restrictions if the images provided are of low resolution or if environmental factors cause variations in image quality.
2. The application is limited to rice leaf diseases. It does not cover other possible problems that could impact rice crops like soil health or pest infestations.
3. If the model encounters previously unseen variations or there is insufficient data for certain disease types, the accuracy of the application may be limited.
4. The limitations on offline functionality will affect the availability of disease detection services in real time.

1.5 Report Organization

This report's first chapter presents a general introduction to the system, outlining its objectives, scope, and limitations.

The subsequent chapters are organized as follows: Chapter 2 provides a review of related works, along with a thorough analysis of project requirements and feasibility. Chapter 3 outlines the system design and describes the basic process model. Implementation and testing of the system are covered in Chapter 4. Finally, Chapter 5 delves into the conclusions drawn from the study.

CHAPTER 2

REQUIREMENT AND FEASIBILITY ANALYSIS

2.1 Literature Review

Numerous studies have already been conducted to identify the diseases found in rice cultivation, and various outcomes have been attained by using various methodologies.

The paper [2] proposed a CNN architecture based on VGG-16 which was trained and tested on a small dataset collected from rice fields and the internet. The dataset consisted of 1649 images of diseased leaves of rice consisting of three most common diseases namely Rice Leaf Blast, Rice Leaf Blight, and Brown Spot and 507 images of Healthy leaves. The accuracy achieved by the proposed model was 92.46%.

The paper [3] used Gaussian Naïve Bayes classifier to categorize the disease into different groups based on the percentage of RGB value of the affected area on a small dataset. Three rice diseases, rice brown spot, rice bacterial blight, and rice blast, have been successfully detected and identified using this method with accuracy above 90%.

Similarly, the paper [4] aimed to use Deep Learning techniques to quickly and easily identify plant diseases where they trained their model with 30880 original images using the Caffe deep learning framework, and tested it using 2589 original images. The authors applied the 10-fold cross validation technique to their dataset to increase evaluation accuracy for predictive models which helped them to achieve accuracy of 96.77%.

The authors of [5] employed machine learning (ML) and image processing methods for the identification and categorization of diseases affecting rice plants. They implemented K-means clustering for image segmentation of rice leaf disease and SVM for classification and got accuracy of 93.33% for training data and 73.33% for testing data.

2.2 Requirements Analysis

This involves the systematic examination and understanding of needs and expectations to define what the system must accomplish and to create detailed specification of system's functional and non-functional requirements. Requirements will be collected through brainstorming, Google searches, feasibility studies, website visits and friendly suggestions.

2.2.1 Functional Requirement

A system needs to be able to perform a functional requirement. It includes descriptions of the work flows carried out by the system, the tasks completed by particular screens, and other business and compliance requirements that the system must fulfill.

- This system should allow users to register and login securely.
- This system should allow users to upload images of rice leaves affected by diseases.
- The machine learning model should be seamlessly integrated into the application for disease classification.
- Users should receive disease classification results promptly.
- The application should also offer guides and information related to rice cultivations.
- The application should be compatible with various web browsers.

2.2.1.1 Use Case Diagram

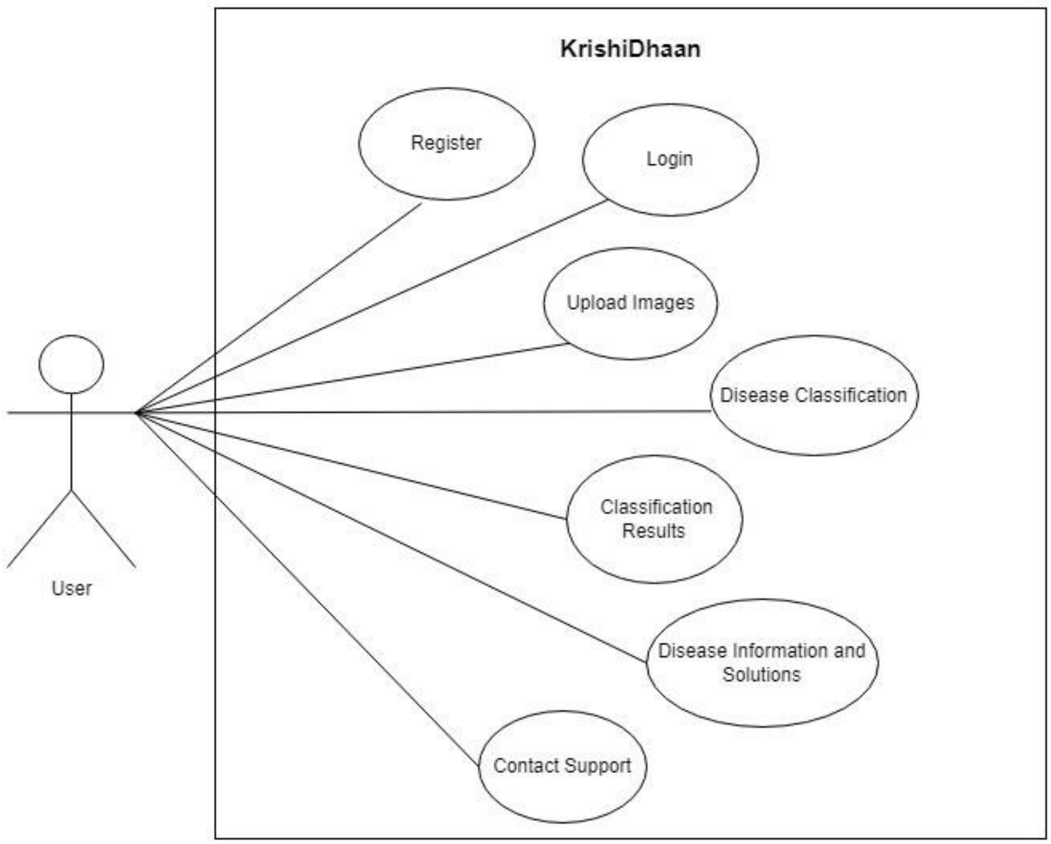

Figure 1: Use Case Diagram for Users

2.2.2 Non-functional Requirement

2.2.2.1 Availability

The system should be available 24/7 for all the users from any geographical location via any platforms.

2.2.2.2 Performance

The system should identify disease with minimal latency by handling multiple concurrent user requests efficiently to provide fast response.

2.2.2.3 Security

The system should implement robust security measures to protect user data and prevent unauthorized access following data protection regulations and best practices.

2.2.2.4 Reliability

The system should be reliable as it identifies disease with high accuracy without any interruption and downtime.

2.2.2.5 Interoperability

The system should be accessible from any devices and should be able to interact with other devices also using standard technologies.

2.3 Feasibility Analysis

It is a thorough evaluation of how well the system will perform within the specified parameters. It investigates the ease of building a system under specified limitations and looks at a number of variables to see if the project can be completed successfully under specified limitations. The limitations encompass the areas of operational, economic, and technical feasibility.

2.3.1 Technical Feasibility

This system's frontend module can be constructed with the help of libraries and technologies like CSS and ReactJS. While the machine learning model for identifying the disease can be developed using open-source machine learning frameworks like TensorFlow and Keras, the backend API modules can be developed using the Django REST Framework. Also, for reading and processing images OpenCV library and for storing information MySQL database can be used which is also freely available. Every technology is available for free and is open-source. It is therefore technically possible.

2.3.2 Operational Feasibility

The proposed system can be accessed from anywhere via Internet using a web browser which is available in both desktop computers and mobile phones. Also, the user guide and support information available in the system will help in the usage of application for all types of users. So, it is operationally feasible.

2.3.3 Economic Feasibility

The system is economically feasible as well because it will be built using the freely available resources from the internet and open-source technologies.

2.4 Software and Hardware Requirement

2.4.1 Software Requirement

Following are the necessary software requirements for this project:

- Python Programming Language
- JavaScript
- ReactJs
- Visual Studio Code
- Postman
- MySQL Relational Database
- Windows/UNIX OS (Operating System)

2.4.2 Hardware Requirement

Following are the necessary hardware requirements for this project:

- Computer Device (Desktop/Laptop)
- Internet Connectivity

2.5 Structuring System Requirements

2.5.1 SRS Document

An SRS document, or Software Requirement Specification, is the blueprint for building software. It outlines what the software should do, how it should work, and who it should be for, much like a map. It includes information on features, user expectations, technical specs, and non-functional needs like security and performance.

2.5.2 Process Modeling

2.5.2.1 Context Diagram

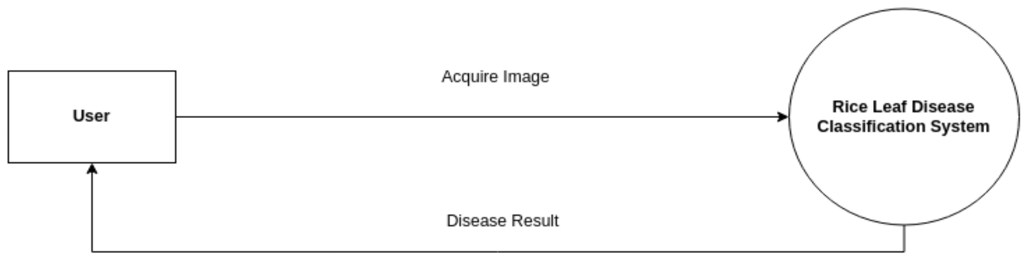

Figure 2: Context Diagram of Rice Leaf Disease Detection System

2.5.2.2 Activity Diagram

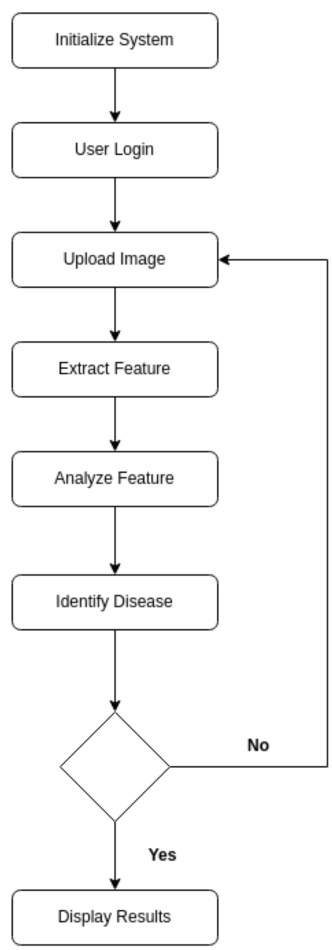

Figure 3: Activity Diagram of Rice Leaf Disease Detection System

CHAPTER 3

SYSTEM DESIGN

3.1 System Architecture

The rice leaf detection and classification is developed as a web-based application which incorporates a deep learning model like CNN in the backend to classify the disease. It provides a simple and user-friendly interface which allows users to upload the images of affected leaves and get early diagnosis of the plant. Additionally, the application displays comprehensive data about disease and the steps that can be taken to prevent or manage them. Only the authorized user will be able to access the service after successfully logging into the system. When the user uploads an image for disease detection, it is transmitted via an API endpoint to the backend server where it is pre-processed—that is, resized and rescaled—to make it suitable for the machine learning model before being fed into the model for prediction. The web server returns the classified disease and its confidence percentage as a result back to the user via API and is displayed in the interface of the system.

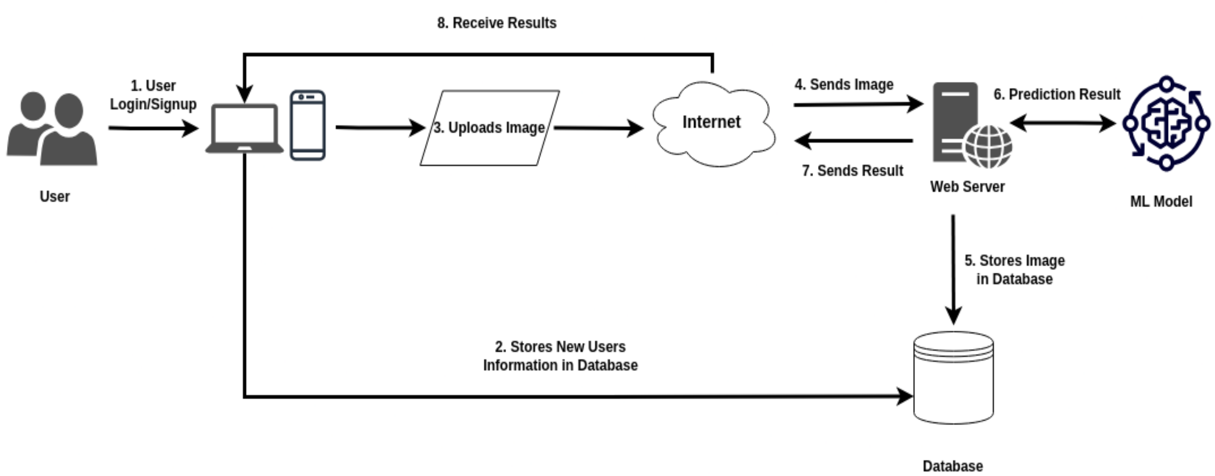

Figure 4: Overall System Architecture of Rice Leaf Disease Detection System

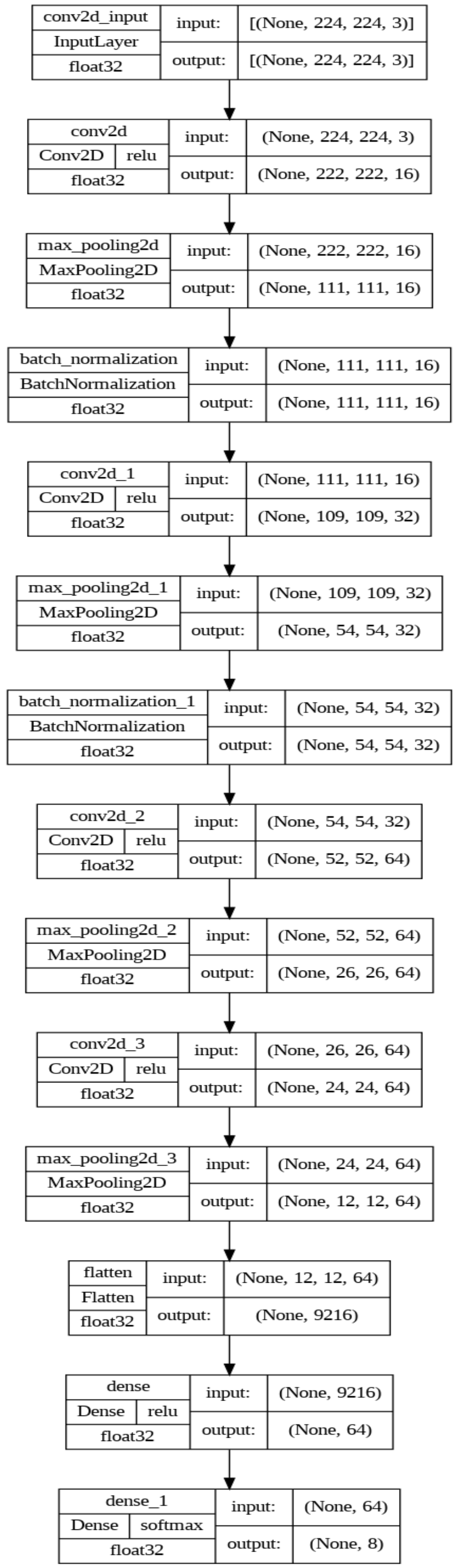

Figure 5: Architecture of CNN Model

3.2 Process Design

The process designs of the system are shown below:

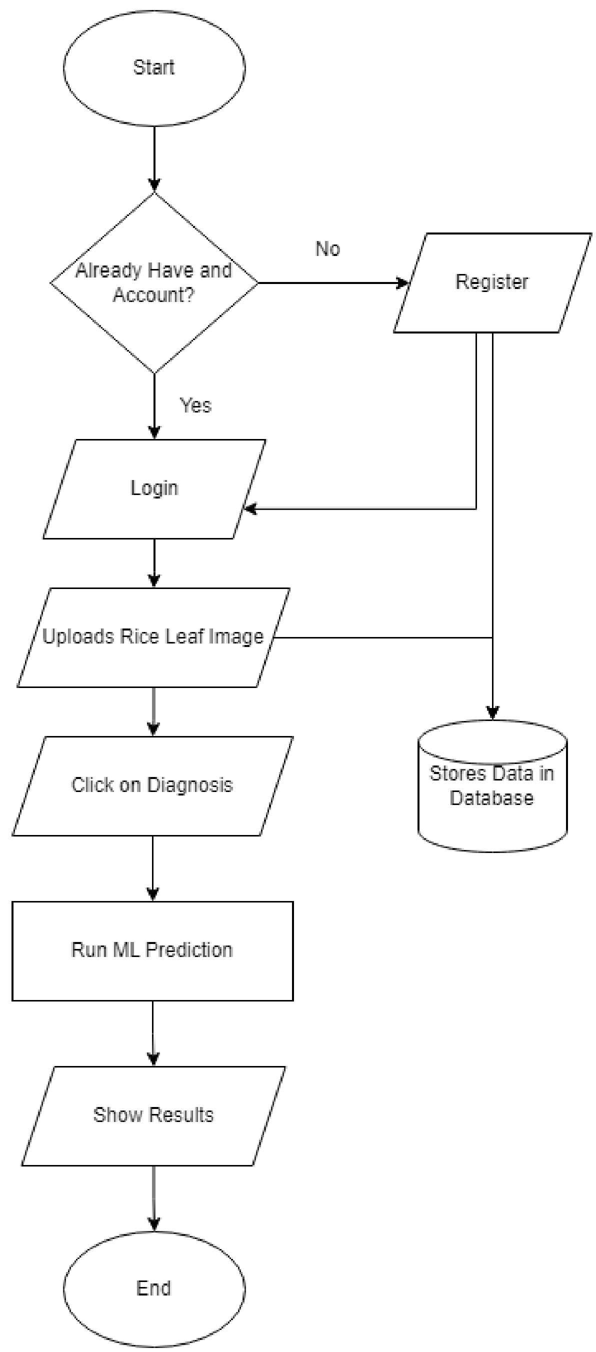

Figure 6: Flowchart for Users

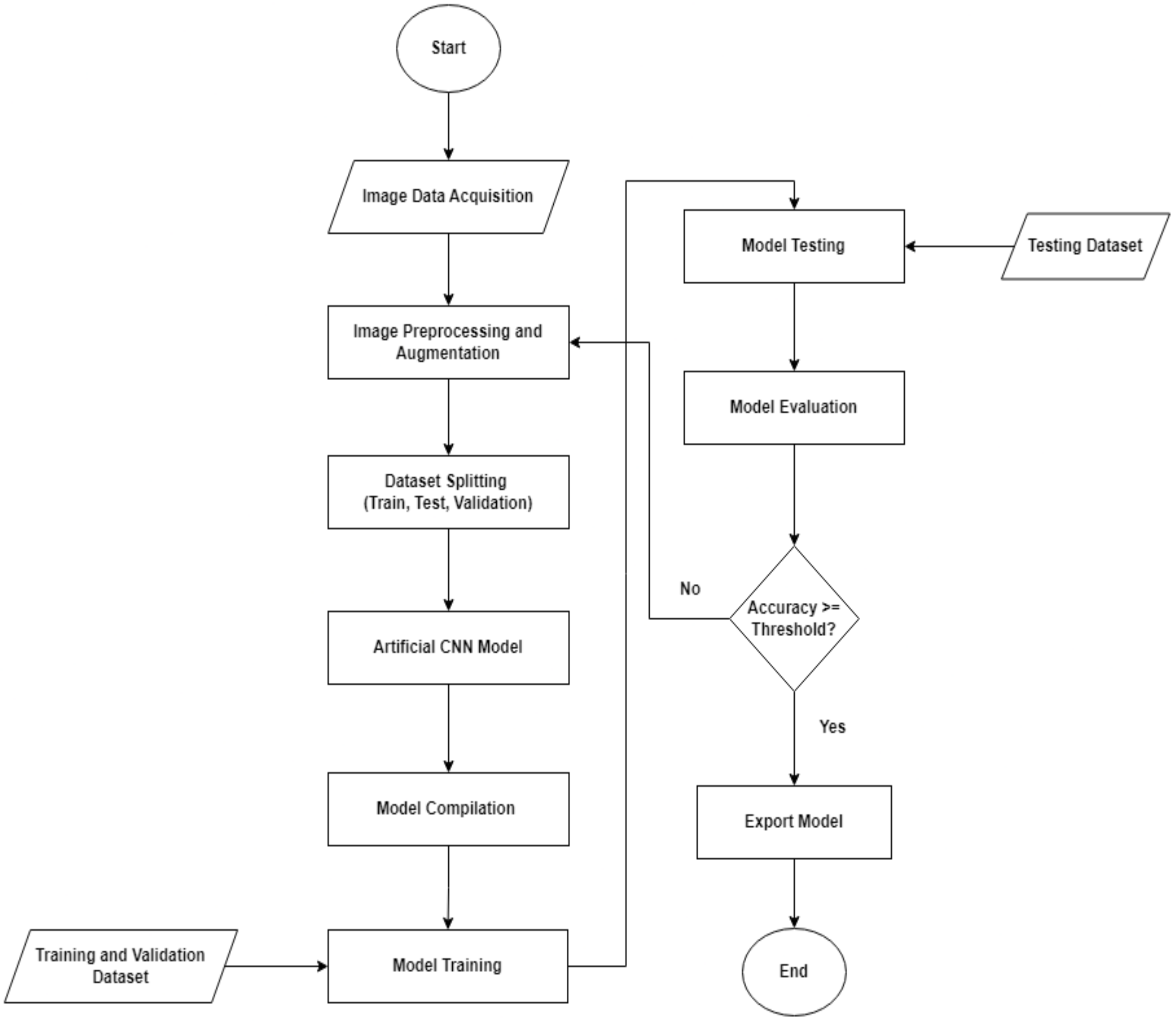

Figure 7: Flowchart for ML Model

3.3 Interface Design

User Interface is the point of interaction between a user and computer system or software application and includes all the tactile, and visual components that enables users to interact with and control the application. The user interfaces for the application can be seen as below:

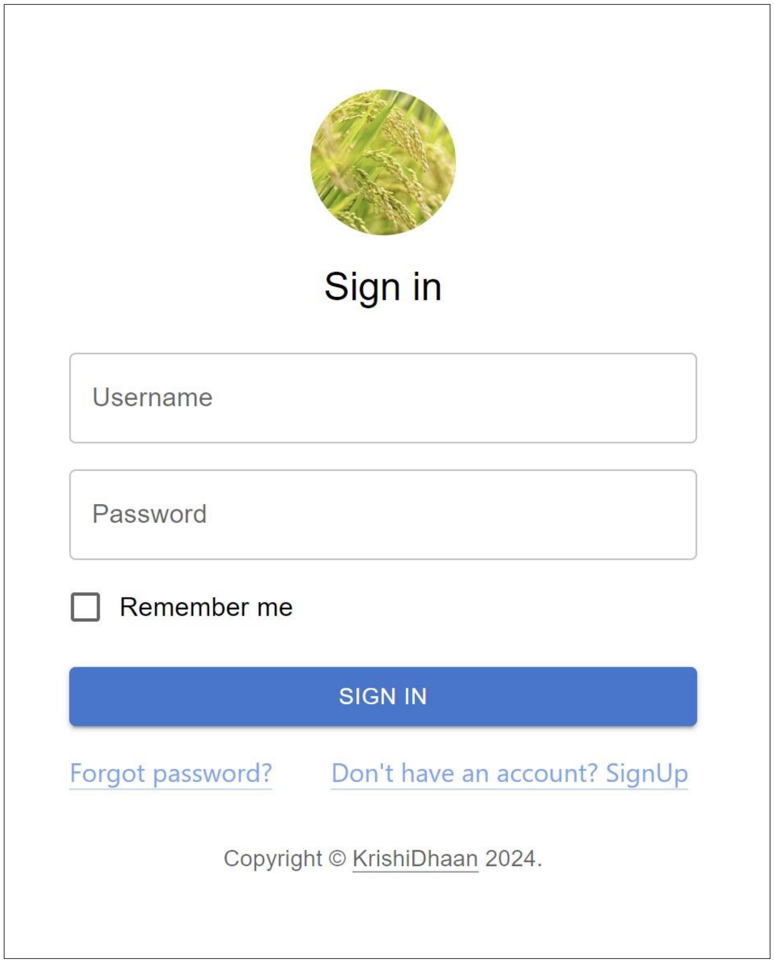

Figure 8: Sign in User Interface

Figure 9: Landing Page Interface

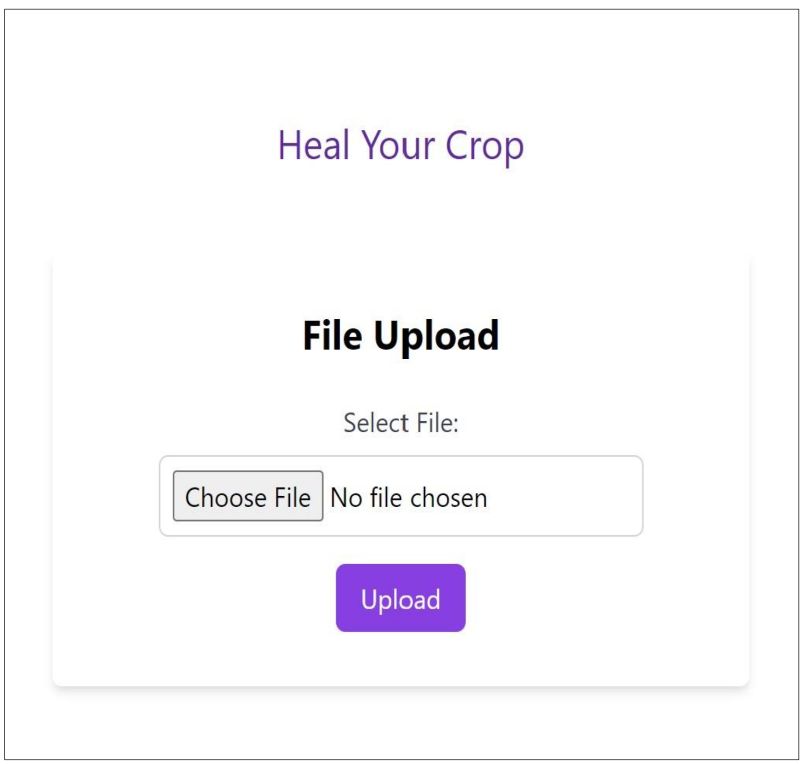

Figure 10: Image Upload Interface

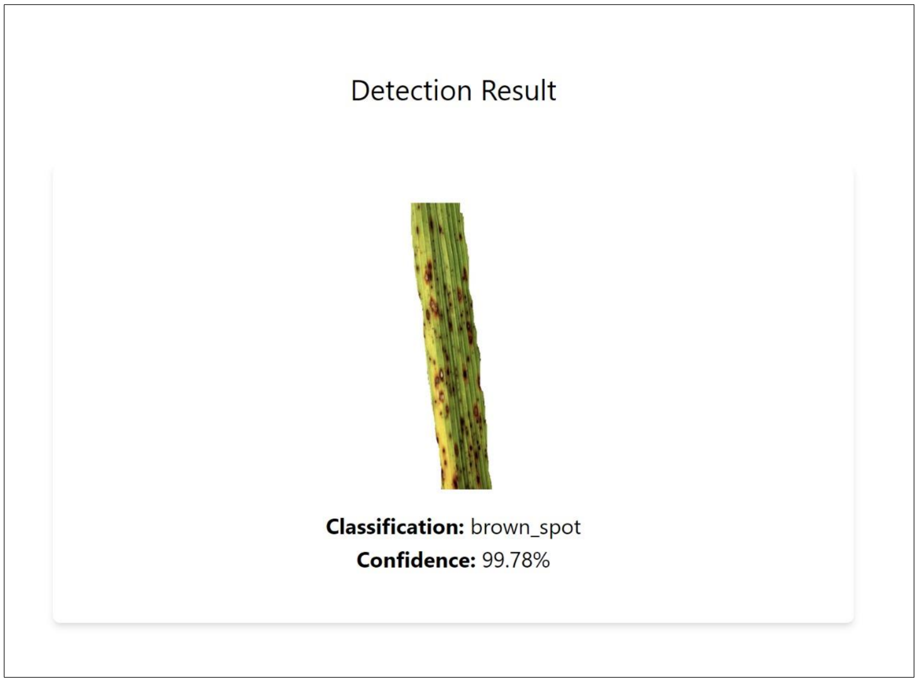

Figure 11: Detection Result Interface

CHAPTER 4

IMPLEMENTATION AND TESTING

4.1 Implementation

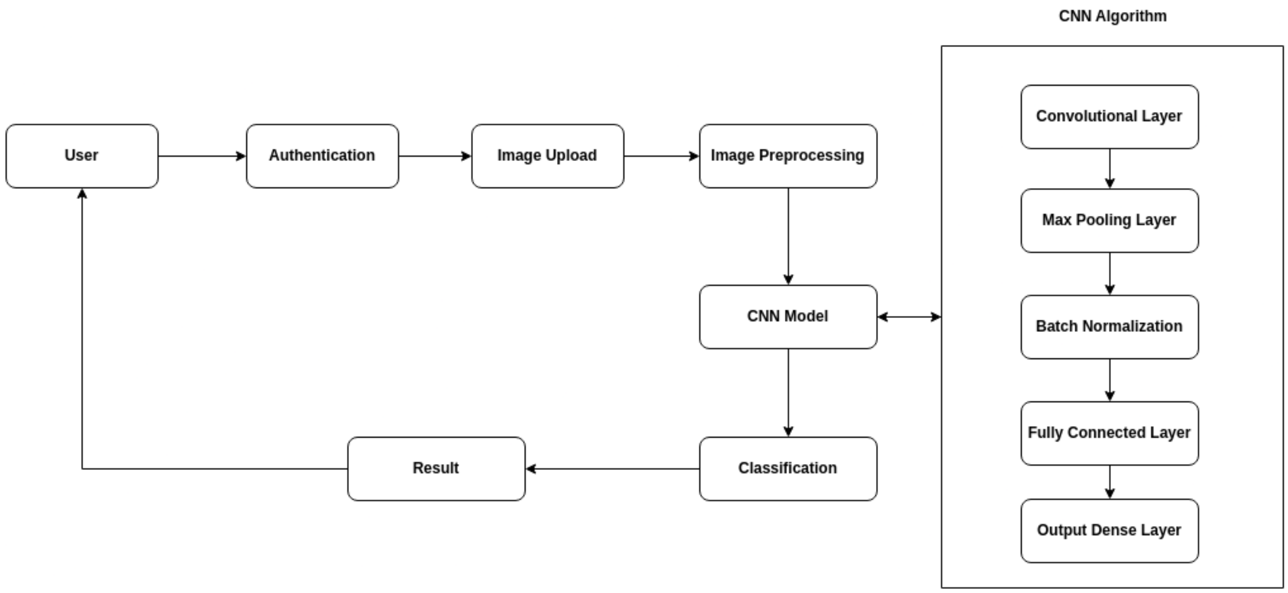

Figure 12: Implementation of Rice Leaf Disease Detection System

To guarantee a reliable and user-friendly experience, the Rice Leaf Disease Detection System implementation process, which is a web-based application, complies with a systematic and sequential flow. Users access the web application and engage with it to start the process. An authentication step is included, requiring users to authenticate before continuing, to ensure secure access and data integrity. It then offers users the ability to upload images of rice leaves for examination. After uploading, the system goes through a preprocessing stage, during which the photos are improved and made ready for feature extraction. Based on the features that were extracted, the Convolutional Neural Network (CNN) model—a crucial part of the system—is then used to classify diseases. The algorithm then produces results that include details about the diseases that have been identified after classification. Ultimately, the workflow is concluded when these outcomes are smoothly returned to the user. The Rice Leaf Disease Detection System operates smoothly and efficiently, giving users insightful information about the condition of their rice crops.

4.2 Algorithm Used

For the purpose of reliably and quickly identifying the rice leaf disease from an uploaded image, the system will employ a state-of-the-art technique that blends transfer learning with the Convolutional Neural Network (CNN) algorithm.

4.2.1 Convolutional Neural Network

It is a type of deep learning model that is specifically designed for tasks involving image processing, recognition and classification. By enabling the machines to automatically learn and extract information from images, convolutional neural networks have revolutionized the field of computer vision. This makes them highly effective in a variety of applications, including image categorization, object detection, and facial recognition.

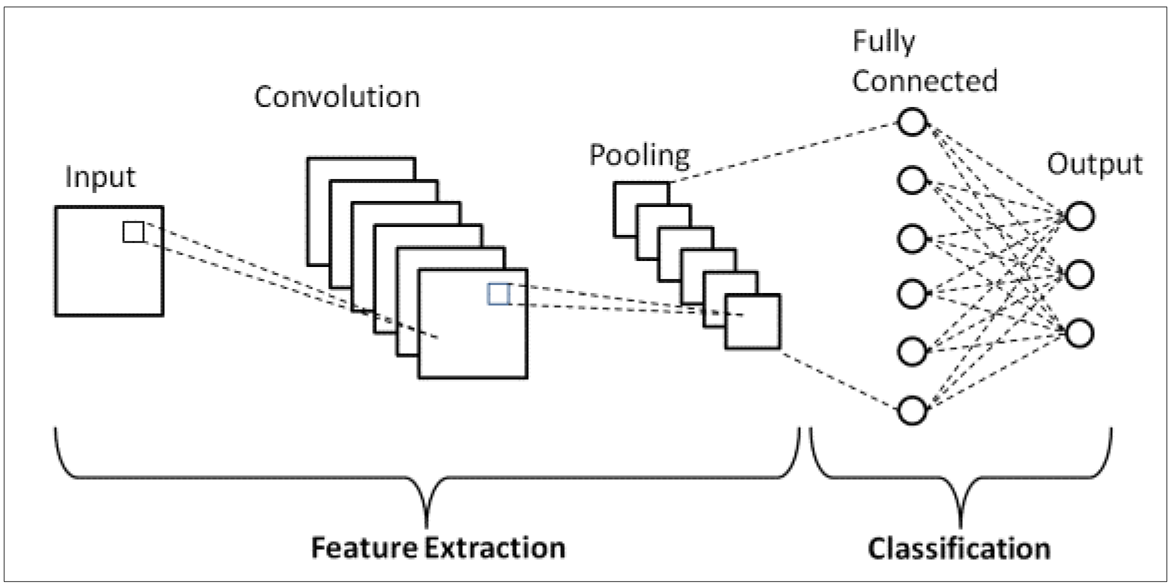

Figure 13: Convolutional Neural Network

Convolutional Neural Network involve several layers and operations, including convolutional layers, pooling layers, fully connected layers and an output layer.

4.2.1.1 Convolutional Layer:

This layer is responsible for applying filters (also known as kernels) to the input images in order to find regional patterns like edges and textures. These filters apply convolutions operation which involves sliding the filter over the entire image, performing this element-wise multiplication and summation at each position to generate feature maps.

Let's say, we have some input image of size $N \times N$ and we use an $m \times m$ filter ω, then our convolutional layer output will be of size $(N-m+1) \times (N-m+1)$ after applying convolution operation. [7]

4.2.1.2 Pooling Layer:

This layer down-sample the feature maps generated by convolutional layer to reduce the spatial dimensions. Max-pooling and average-pooling are frequent pooling operations that preserve crucial data while minimizing computational complexity. It simply takes a $k \times k$ region and output a single value. [7]

4.2.1.3 Batch Normalization Layer

The batch normalization layer is responsible for addressing challenges related to internal covariate shift which refers to the change in the distribution of input values to a neural network's layers during training, which can impede convergence and slow down the learning process. It involves normalizing the input of a layer by subtracting the mean and dividing it by the standard deviation. The normalized values are then scaled and shifted using learnable parameters.

For a input feature x in a mini-batch, the Batch Normalization operation is expressed as follows:

$$\boxed{\text{Batch Normalization}(x) = \gamma . \frac{x-\mu}{\sigma} + \beta}$$

Where,

x is the input feature for a specific neuron in a layers,

μ is the mean of the mini-batch, calculated as the average of the feature values in the mini-batch,

σ is the standard deviation of the mini-batch calculated as the square root of the variance of the feature values in the mini-batch,

γ is the scale parameter, a learnable parameter that scales the normalized values,

β is the shift parameter, a learnable parameter that shifts the scaled and normalized values.

4.2.1.4 Fully Connected Layer:

This layer is responsible for connecting every neuron to every neuron to the previous layer. They make high-level decisions and use reasoning based on features taken from earlier layers.

4.2.1.5 Output Layer:

This layer produces the final prediction. The number of neurons in this layer corresponds to the number of classes in a classification task. Activation function like ‘softmax’ is used for multi-class classification.

4.2.2 Transfer Learning

It is a common machine learning strategy that involves applying a model that has already been trained to a related task in order to take advantage of its prior experience and increase efficiency and effectiveness. Pre-trained models can be fine-tuned for a target task, reducing data requirements, training time, and frequently improving performance because they have already learned useful features from a source task. This method is widely used in computer vision, natural language processing, speech recognition, and recommendation systems.

For this proposed system, pre-trained models like VGG (Visual Geometry Group), ResNet (Residual Network), AlexNet or Inception Model can be used to extract the features from images and custom layers can be further added which is specific to our task. Also, the combination of models like Inception-ResNet model. These models contains multiple convolutional layers with different filters (kernels) densely connected with each other allowing to capture fine-grained features at different scales.

7 References

[1] K. Joshi, S. Upadhyay, P. Chaudhary, S. Shrestha, K. Bhattarai and B. Tripathi, "The Rice Processing Industry in Nepal: Constraints and Opportunities," *Agricultural Sciences,* 2020.

[2] S. Ghosal and K. Sarkar, "Rice Leaf Diseases Classification Using CNN With Transfer Learning," IEEE Calcutta Conference (CALCON), Kolkata, 2020.

[3] T. Islam, M. Sah, S. Baral and R. Roy Choudhury, "A Faster Technique on Rice Disease Detectionusing Image Processing of Affected Area in Agro-Field," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, 2018.

[4] M. Van Hulle, S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk and D. Stefanovic, " Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification," Hindawi Publishing Corporation, 2016.

[5] H. Prajapati, J. Shah and V. Dabhi, "Detection and Classification of Rice Plant Diseases.," Intelligent Decision Technologies 11 (2017) 357–373, 2017.

[6] TutorialsPoint, "SDLC-Waterfall Model," TutorialsPoint, [Online]. Available: https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm. [Accessed 7 October 2023].

[7] A. Gibiansky, "Andrew Gibiansky :: Math → [Code]," 24 February 2014. [Online]. Available: https://andrew.gibiansky.com/blog/machine-learning/convolutional-neural-networks/. [Accessed 7 October 2023].