**CHAPTER-1**

**INTRODUCTION**

Sugarcane is a vital crop for the agricultural industry, and Pakistan is one of the major producers and exporters of sugarcane. However, sugarcane diseases can cause significant damage to crops, resulting in huge financial losses for farmers. Therefore, it is essential to develop an accurate and efficient system to predict and manage sugarcane diseases. In recent years, there has been a growing interest in using machine learning and image processing techniques for crop disease detection. This project aims to contribute to the existing literature on using these techniques by developing an automated web application that utilizes machine learning and image processing to predict the presence of disease in sugarcane crops and recommend treatments. The system will use digital images of sugarcane leaves and stems, and through image processing techniques, the system will extract features that can be used to identify the type of disease present in the crop. Machine learning algorithms will then be trained on these features to accurately predict the presence of disease. Once the disease is identified, the system will recommend appropriate treatments to manage its effects and improve crop yield. The recommended treatments will be based on scientific research and will be tailored to the specific type of disease present in the crop. The proposed system has the potential to revolutionize the way sugarcane diseases are managed by providing farmers with a reliable and efficient tool to identify and manage diseases, reduce crop losses, and improve the sustainability of the sugarcane industry.

* 1. **Project Overview:**

Sugarcane is a significant cash crop in Pakistan, but it is vulnerable to heavy losses caused by over 50 different diseases[1], including fungi, bacteria, viruses, and nematodes. These diseases can occur in certain areas, seasons, and specific parts of the plant, making them difficult to identify and manage. Therefore, the development of an automated system for disease prediction and treatment is crucial to mitigate crop losses and improve crop yield. Furthermore, managing sugarcane diseases can be a challenging task for farmers due to the complex nature of the sugarcane plant and the diverse range of diseases that can affect it. Sugarcane diseases can lead to significant economic losses, impacting the livelihoods of farmers and the overall sustainability of the sugarcane industry. Therefore, the need for an automated system that can accurately predict and manage sugarcane diseases has become increasingly important. By utilizing machine learning and image processing techniques, an automated system can provide farmers with a reliable and efficient tool to detect and manage diseases, reducing crop losses and improving the overall sustainability of the sugarcane

* 1. **Objectives:**

The proposed project on automated sugarcane disease prediction and treatment is aimed at providing farmers with a valuable tool to help manage sugarcane diseases, reduce crop losses and improve the sustainability of the sugarcane industry. The project objectives include the accurate identification of the type of disease present in sugarcane crops and the provision of recommended treatment options to mitigate its effects and improve crop yield. These objectives will be achieved using machine learning and image processing techniques, which will enable the system to accurately identify the disease and provide recommended treatment options. Disease identification is a critical component of the project, as sugarcane crops can be affected by various types of diseases that can impact their quality and quantity. By utilizing machine learning and image processing techniques, the system will be able to accurately identify the disease present in the sugarcane crops, thus providing farmers with an efficient way to manage the disease. In addition to disease identification, the system will also provide recommended treatment options to mitigate the effects of the disease and improve crop yield. These recommendations will be based on the type of disease identified, as different diseases may require different treatment options. The provision of recommended treatment options will enable farmers to take proactive measures to manage the disease, thus reducing crop losses and improving the sustainability of the sugarcane industry.

#### 1.3Project scope:

Early detection of the sugarcane disease may enable the farmer to control its spread before attempting to treat the damaged plant. Below are some examples of in- and out-of-scope topics.

#### 1.3.1 In scope:

* Disease prediction: The project aims to develop a system that can accurately predict the presence of diseases in sugarcane plants using machine learning and image processing techniques.
* Treatment suggestion: Once a disease is identified, the system should provide recommended treatment options to help mitigate the effects of the disease and improve crop yield. These recommendations may include the use of pesticides or other disease control measures.

#### 1.3.2Out of scope:

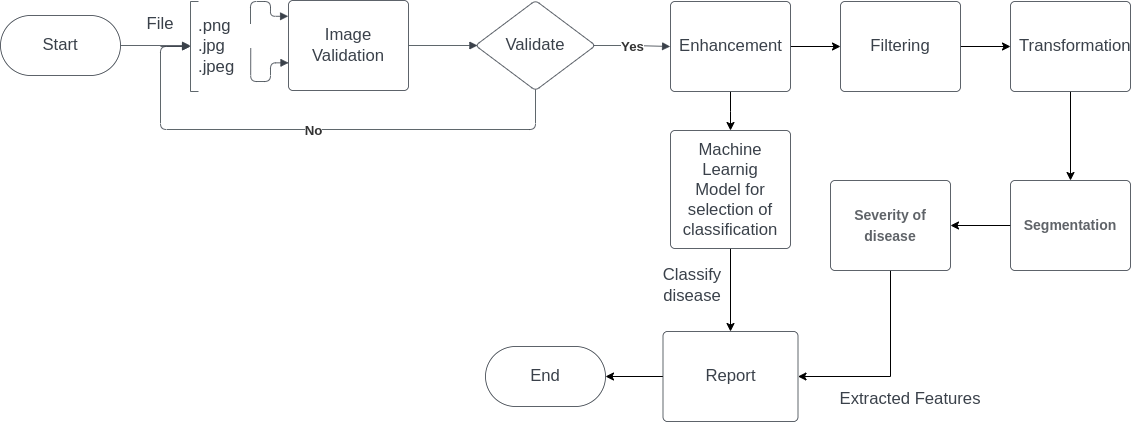
* Implementing the treatment: While the project may provide suggestions for treatment, it is not responsible for actually implementing the treatment. The implementation would be the responsibility of the farmer or the relevant authorities.
* Monitoring the crop after treatment: While the system may suggest treatment options, it does not include monitoring the crop after the treatment to assess its effectiveness.
* Disease prevention: While the early detection of the disease can help in preventing its spread, the project is mainly focused on the detection and treatment of the disease rather than its prevention.
* Cost analysis: The project is not focused on the cost analysis of the treatment options suggested, this aspect should be considered separately.
  1. **System Diagram**:

Figure No. 1 System Diagram The process of developing an automated sugarcane disease predictThe process of developing an automated sugarcane disease prediction and treatment system involves several key steps. The first step is to ensure that the system can correctly identify and classify images of sugarcane crops. This is accomplished through the use of advanced image processing techniques that can distinguish sugarcane images from other types of images. Once the system confirms that the uploaded image is an image of sugarcane, a machine learning model is used to extract relevant features from the image. These features can include characteristics such as color, texture, and shape, which are used to identify the disease present in the image. The machine learning algorithm is trained on a large dataset of sugarcane disease images to accurately classify new images. After identifying the disease, the system provides recommendations for treatment. The treatment recommendations are based on the disease's severity and can range from simple measures such as crop rotation or pruning to more intensive treatments such as pesticide applications or biological control methods. The automated sugarcane disease prediction and treatment system provides a valuable tool for farmers to manage sugarcane diseases more efficiently. With accurate disease identification and treatment recommendations, farmers can reduce crop losses and improve the sustainability of the sugarcane industry. Additionally, the system can help inexperienced individuals in the field of horticulture to detect sugarcane diseases quickly and effectively. Overall, the system represents a significant advancement in the field of precision agriculture and has the potential to revolutionize sugarcane crop management.

**CHAPTER-2**

**LITERATURE SURVEY**

"An Upgraded Strategy for Division and Order of Apple Illnesses Dependent on Solid Connection and Hereditary Calculation Based Component Choice" Apple Infection ID was executed here [9]. "An Information Expansion Technique Dependent on Generative Ill-disposed Organizations for Grape Leaf Infection Distinguishing proof" The ID of grape leaf illnesses upheld profound learning is crucial for look the unfurl of sicknesses and ensuring the sound development of the grape exchange [10]. ID models [11]. profound lament angle punishment system is applied to settle the model's instructing position amount [12]. Grape leaf unhealthiest film region unit offered here [13]. Grape Leaf distinguishing proof is urgent in agribusiness [14]. "Programmed Acknowledgment of Soybean Leaf Illnesses Utilizing UAV Pictures and Profound Convolutional Neural Organizations" Plant infections are a significant worry in agribusiness [15]. "Bacterial Searching Advancement Based Spiral Premise Capacity Neural Organization (BRBFNN) for Recognizable proof and Arangement of Plant Leaf Illnesses: A Programmed Approach

Convolutional Neural Networkshave an unequivocal style than standard Neural Affiliations. Standard Neural Affiliations Re-Model Associate in Nursing responsibility by swing it through a development of disguised layers. each layer is shaped of a gaggle of neurons, where each layer is totally associated with any or all neurons among the layer as of now. Convolutional Neural Affiliations unit scarcely entire astonishing [25]. right off the bat of the entirety of, the layers unit unionized in 3 assessments: assessment, stature and importance. Further, the neurons in a single layer don't append with the entirety of the neurons among succeeding layer yet to a piece space of it. With everything taken into account, the final word yield goes to be lessened to in any event one vector of chance scores, formed on the importance assessment.Fig-1 Sugarcane Leaf Disease Identification with CNN

CHAPTER-3

## AIM AND STATEMENT OF PROBLEM

#### 3.1Problem:

Sugarcane, being a vital cash crop worldwide, is vulnerable to numerous diseases, with more than 50 different types documented in the literature. Among these, those caused by fungi, bacteria, viruses, and nematodes are considered to be the most devastating. They can occur in specific regions, during specific seasons, and affect specific parts of the plant. These diseases can affect all parts of the sugarcane plant, and they are prevalent in almost every field and on every plant. The symptoms and signs they exhibit are alarming, and their impact on the quality and quantity of sugarcane production is significant. The gravity of the problem highlights the importance of finding effective ways to manage the crop and enhance its yield for researchers, farmers, and all relevant stakeholders.

The unpredictability and complex nature of sugarcane diseases make their early detection and treatment challenging, and this can lead to heavy losses for farmers. Traditional methods of disease detection and management are time-consuming and require expert knowledge, making them less accessible for many farmers. This situation calls for the development of an automated system that can assist in the prediction and treatment of sugarcane diseases, making it more accessible for farmers to manage the crop and protect it from significant losses.

Recent research has shown that the use of machine learning and image processing techniques can significantly improve the accuracy and speed of sugarcane disease detection and treatment. Machine learning algorithms, such as decision trees, support vector machines, and deep learning techniques such as convolutional neural networks (CNNs), have been utilized in various studies to classify sugarcane diseases. Image processing techniques, such as leaf segmentation, feature extraction, and classification, have also been proposed for disease detection, which helps in isolating the leaf from the image, identifying features of the leaf that can be used to identify the disease, and identifying the disease based on the extracted features.

The proposed project aims to develop an automated system that utilizes machine learning and image processing techniques to assist farmers in managing sugarcane diseases and protecting their crops from heavy losses. By utilizing this approach, the proposed system has the potential to automate the process of disease identification and treatment, making it more accessible for farmers. This project will contribute to the existing literature by providing a valuable tool for farmers to manage diseases and protect their sugarcane crops, which will enhance the yield and ensure a sustainable sugarcane industry.

#### 3.2Solution:

The proposed solution to the problem of sugarcane diseases is a timely and necessary innovation that seeks to alleviate the challenges faced by farmers in managing and controlling sugarcane diseases. With over 7 different types of diseases affecting sugarcane plants, the development of an automated disease detection system is critical to ensuring the continued sustainability of the sugarcane industry. By leveraging image analysis and machine learning techniques, the system offers farmers an accessible and reliable tool that can assist in identifying and treating plant infections.

The system's approach to disease detection is particularly innovative, as it analyzes visual symptoms and physical appearance of sugarcane plants to identify potential diseases. This approach is especially useful for those who lack experience in horticulture or are unfamiliar with sugarcane diseases. Moreover, the system's ability to generate a detailed report on the detected disease along with suggested treatment plans will further assist farmers in managing the disease and preventing heavy losses in crops.

The implementation of the proposed solution would provide substantial benefits to the sugarcane industry as a whole. By reducing crop losses and increasing overall crop yield, the solution would improve the economic sustainability of sugarcane production. Furthermore, the solution would help to ensure the continued provision of sugarcane-based products that are essential to various industries, including food and beverage, biofuel, and pharmaceuticalsIn conclusion, the proposed solution is a vital innovation that has the potential to revolutionize the management and control of sugarcane diseases. By leveraging image analysis and machine learning techniques, the system will provide farmers with a valuable tool that can assist in identifying and treating plant infections, ultimately improving the sustainability of the sugarcane industry.

3.2Process:

The Five Main Steps in Developing our model are:

1. Collecting Huge Data: The first step in developing our model is to collect a large amount of data. This data should include a variety of images of sugarcane plants, both healthy and infected with different types of diseases.
2. Image Processing: Once we have collected our data, the next step is to process the images using various image processing techniques. This will include tasks such as image segmentation, feature extraction, and data pre-processing.
3. Train and Test the Dataset: After processing the images, the next step is to train and test our dataset using various machine learning algorithms. This will allow us to evaluate the performance of the model and optimize its parameters.
4. Classification of Diseases and recommendations for treatments: The next step is to use the trained model to classify the diseases present in the images, and provide recommendations for treatments.

The five steps mentioned above are the fundamental steps to develop the model, but there can be variations in the details of the implementation according to the specific problem or data.

**CHAPTER-4**

**SYSTEM REQUIREMENTS**

**4.1 Hardware Requirements**:

The hardware requirements for our automated sugarcane disease prediction and treatment project are: 1. Camera: The system utilizes images of sugarcane plants for analysis, therefore a camera is needed to capture these images. A mobile camera is recommended as it allows for easy data collection in the field. 2. Computer for Image Processing: The captured images will then need to be processed using image processing techniques, this will include tasks such as image segmentation, feature extraction, and data pre-processing. A computer with enough processing power, memory, and storage is needed to run these operations. The computer should be equipped with software development tools (python, libraries, etc), and enough computational power to run Machine Learning models. Additional hardware that could be required depend on the scope and complexity of the project, but the two mentioned above are the minimum requirement.

**4.2 Software Requirements**:

The software requirements for our automated sugarcane disease prediction and treatment project are:

1. **Python**:

Python is a powerful, open-source programming language that is widely used in data science, machine learning, and image processing. It is the main programming language for this project, as it provides a vast ecosystem of libraries, modules and frameworks. Some of the common used libraries are numpy, pandas, opencv, sklearn, and TensorFlow/Pytorch to build the image processing and machine learning models.

1. Additional software might be required depending on the specific requirements of the project and complexity, but Python is considered as the backbone of the project. It is important to note that the specific versions of software and libraries used will depend on the particular implementation of the project and the availability of such versions

**3.JUPYTER:**

Jupyter is an open-source web application that allows you to create and share documents containing live code, equations, visualizations, and narrative text. It is commonly used for data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

Anaconda is a distribution of Python and R for scientific computing and data science. It simplifies package management and deployment. Anaconda comes with many useful libraries and tools pre-installed, including Jupyter.

### 4. MATLAB:

MATLAB® combines a desktop environment tuned for iterative analysis and design processes with a programming language that expresses matrix and array mathematics directly. It includes the [Live Editor](https://in.mathworks.com/products/matlab/live-editor.html) for creating scripts that combine code, output, and formatted text in an executable notebook.

## 5.Tensor Flow Network:

Tensor flow is an open-source software library. Tensor flow was originally developed by researchers and engineers. It is working on the Google Brain Team within Google’s Machine Intelligence research organization the purposes of conducting machine learning and deep neural networks research.

It is an opensource framework to run deep learning and other statistical and predictive analytics workloads.

It is a python library that supports many classification and regression algorithms and more generally deep learning.

**6.OPENCV:**

1.It is a cross-platform library using which we can develop real-time computer vision applications

Fig.4.1: OpenCV

It mainly focuses on image processing, video capture and analysis including feature likeface detection and object detection.

Currently Open CV supports a wide variety of programming languages like C++, Python,Java etc. and is available on different platforms including Windows, Linux, OS X, Android,iOS etc. Also, interfaces based on CUDA and OpenCL are also under active development for high- speed GPU operations. Open CV-Python is the Python API of Open CV.

It combines the best qualities of Open CV C++ API and Python language.

OpenCV (Open-Source Computer Vision Library) is an opensource 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. Being a BSD-licensed product, OpenCV makes it easy for businessesto utilize and modify the code.

### 7.NUMPY:



Fig.4.2: NumPy

NumPy is a library for the Python programming language, adding support for large, multi- dimensional arrays and matrices, along with a large collection of highlevel mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Num array into Numeric, with extensive modifications. NumPy is opensource software and has many contributors.

**MATPLOT:**

Mat plot is a plotting library for the Python programming languageand its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, WX Python, Qt, or GTK+. There is also a procedural "Pylab" interface based on a state machine(like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged SciPy makes use of Matplotlib*.*

**CHAPTER-5**

**SYSTEM ANALYSIS**

Analysis is the process of finding the best solution to the problem. System analysis is the process by which we learn about the existing problems, define objects and requirements and evaluates the solutions. It is the way of thinking about the organization and the problem it involves, a set of technologies that helps in solving these problems. Feasibility study plays an important role in system analysis which gives the target for design and development.

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test theTechnical, Operational and Economical feasibility for adding new modules and debuggingold running system. All system is feasible if they are unlimited resources and infinite time.There are aspects in the feasibility study portion of the preliminary investigation:

* + 1. Technical Feasibility
    2. Economical Feasibility
    3. Operation Feasibility

### 5.1 TECHNICAL FEASIBILITY:

In the feasibility study first step is that the organization or company has to decidethat what technologies are suitable to develop by considering existing system.

The technical issue usually raised during the feasibility stage of the investigationincludes the following:

Does the necessary technology exist to do what is suggested?

Do the proposed equipment have the technical capacity to hold the data required to usethe new system?

Will the proposed system provide adequate response to inquiries, regardless of thenumber or location of users?

Can the system be upgraded if developed?

Are there technical guarantees of accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to the users. Thedatabase’s purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified.

Therefore, it provides the technical guarantee of accuracy, reliability and security. The software and hard requirements for the development of this project are not many and are already available in-house at NIC or are available as free as open source. The work for theproject is done with the current equipment and existing software technology. Necessary bandwidth exists for providing a fast feedback to the users irrespective of the number of users using the system.

Here in this application used the technologies like **Visual Studio 2012 and SqlServer** 2014. These are free software that would be downloaded from web.

### 5.2 ECONOMICAL FEASIBILITY

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

The system is economically feasible. It does not require any addition hardware or software.Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility forcertain.

Determining Economic Feasibility:

Assessing the economic feasibility of an implementation by performing a cost/benefit analysis, which as its name suggests compares the full/real costs of the application to its full/real financial benefits. The alternatives should be evaluated on the basis of their contribution to net cash flow, the amount by which the benefits exceed the costs, because the primary objective of all investments is to improve overall organizational performance

|  |  |  |
| --- | --- | --- |
| **Type** | **Potential Costs** | **Potential Benefits** |
| Quantitative | Hardware/softwareupgrades  Fully-burdened cost of labor (salary + benefits)  Support costs for theapplication Expected operationalcosts  Training costs for usersto learn the application  Training costs to traindevelopers in new/updated technologies | Reduced operating costs  Reduced personnel costs from areduction in staff  Increased revenue from additional sales of your organizations products/services |
| Qualitative | Increased employee dissatisfaction from fearof change | Improved decisions as the resultof access to accurate and timelyinformation  Raising of existing, or introduction of a new, barrier toentry within your industry to keep competition out of your market  Positive public perception that your organization is an innovator |

### OPERATIONAL FEASIBILITY:

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of important issues raised are to test the operational feasibility of a project includes the following

Is there sufficient support for the management from the users?

Will the system be used and work properly if it is being developed and implemented?

Will there be any resistance from the user that will undermine the possible applicationbenefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand,the management issues and user requirements have been taken into consideration. So thereis no question of resistance from the users that can undermine the possible application benefits.

|  |  |
| --- | --- |
| **Operations Issues** | **Support Issues** |
| What tools are needed to supportoperations? What skills will operators need to be trainedin? What processes need to be created and/orupdated? What documentation does operations need? | What documentation will users be given?  What training will users be given?  How will change requests be managed? |

**CHAPTER-6**

**SYSTEM DESIGN**

Design is a creative process; a good design is the key to effective system. The system Design is defined as “The process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. Various design features are followed to develop the system. The design specification describes the features of the system, the components or elements of the system and their appearance to end- users.

## 6.1 Fundamental Design Concepts

A set of fundamental design concepts has evolved over the past three decades. Although the degree of interest in each concept has varied over the years, each has stood the test of time. Each provides the software designer with a foundation from which more sophisticated design methods can be applied. The fundamental design concepts provide the necessary framework for “getting it right”. The fundamental design concepts such as abstraction, refinement, modularity, software architecture, control hierarchy, structural partitioning, data structure, software procedure and information hiding are applied in this project to getting it right as per the specification.

### Input Design

The input Design is the process of converting the user-oriented inputs in to the computer-based format. The goal of designing input data is to make the automation as easy and free from errors as possible. Providing a good input design for the application easy data input and selection features are adopted. The input design requirements such as user friendliness, consistent format and interactive dialogue for giving the right message and help for the user at right time are also considered for the development of the project. Input design is a part of overall system design which requires very careful attention. Often the collection of input data is the most expensive part of the system, which needs to be route through number of modules.

### 6.1.2 Output Design

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other systems through outputs. It is most important and direct source information to the user. Efficient and intelligent output improves the systems relationship with source and destination machine.

## System development methodology

System development method is a process through which a product will get completed or a product gets rid from any problem. Software development process is described as a number of phases, procedures and steps that gives the complete software. It follows series of steps which is used for product progress. The development method followed in this project is waterfall model.

### Model phases

The waterfall model is a [sequential](http://en.wikipedia.org/wiki/Sequence) software development process, in which progress is seen as flowing steadily downwards (like a [waterfall](http://en.wikipedia.org/wiki/Waterfall)) through the phases of Requirement initiation, [Analysis](http://en.wikipedia.org/wiki/Analysis), [Design,](http://en.wikipedia.org/wiki/Design) Implementation, [Testing](http://en.wikipedia.org/wiki/Software_testing) and [maintenance.](http://en.wikipedia.org/wiki/Software_maintenance)

**Requirement Analysis:** This phase is concerned about collection of requirement of the system. This process involves generating document and requirement review.

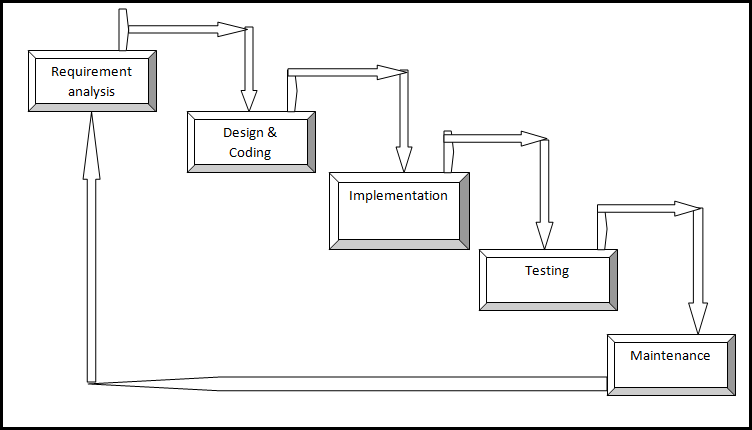
**System Design:** Keeping the requirements in mind the system specifications are translated in to a software representation. In this phase the designer emphasizes on:-algorithm**,** data structure**,** software architecture etc.

**Coding:** In this phase programmer starts his coding in order to give a full sketch of product. In other words system specifications are only converted in to machine readable compute code.

**Implementation:** The implementation phase involves the actual coding or programming of the software. The output of this phase is typically the library, executables, user manuals and additional software documentation

**Coding:** In this phase programmer starts his coding in order to give a full sketch of product. In other words system specifications are only converted in to machine readable compute code.

**Implementation:** The implementation phase involves the actual coding or programming of the software. The output of this phase is typically the library, executables, user manuals and additional software documentation



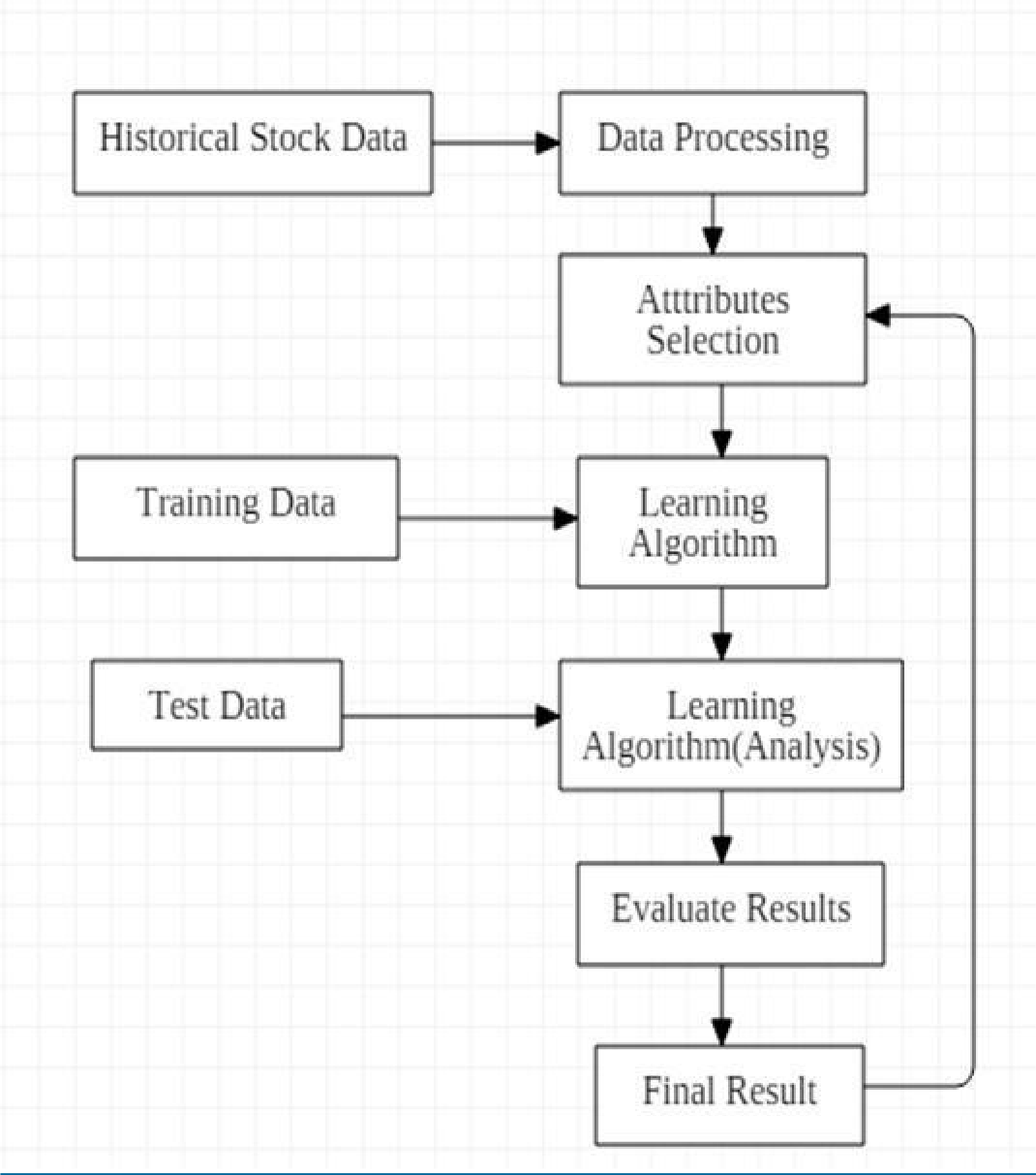
**6.2.2 Reason for choosing waterfall model as development method**

### Fig.3:- Waterfall model

* + - * Clear project objectives.
      * Stable project requirements.
      * Progress of system is measurable.
      * Strict sign-off requirements.
      * Helps you to be perfect.
      * Logic of software development is clearly understood.
      * Production of a formal specification
      * Better resource allocation.
      * Improves quality. The emphasis on requirements and design before writing
      * a single line of code ensures minimal wastage of time and effort and reduces the risk of schedule slippage.
      * Less human resources required as once one phase is finished those people can start working on to the next phase.

**CHAPTER-7**

**SYSTEM ARCHITECTURE**

System architecture is the conceptual design that defines the [structure](http://en.wikipedia.org/wiki/Structure) and [behavior](http://en.wikipedia.org/wiki/Behavior) of a [system.](http://en.wikipedia.org/wiki/System) An architecture description is a formal description of a system, organized in a way that supports reasoning about the structural properties of the system. It defines the [system](http://en.wikipedia.org/wiki/System)

### Fig 7.1 : System Architecture

The main aim of the project is to predict the Stock market price, in this project we required numpy library for working with arrays, and we require pandas library to analyse the data of given dataset, and Matplotlib library is used to plot the output graph and sklearn.model\_selection is used to split our dataset into testing and training dataset, and sklearn will also provide the machine learning model know as **Support vector machine (SVR) Regression** and it will also provide us the radial basis function for implementing several losses, scores and utility functions to measure classification performance

To predict the stock price we have used the machine learning model called as **Support vector machine(SVR) Regression,** which will take the input dataset of at least 10years previous Bitcoin price of any Bitcoin, pandas is used to read the CSV file provided and it will analyse the dataset provided, then we will drop the unwanted data which is present in the dataset so that dataset will be clean and clear for the machine learning model, then we have assigned the number of days to be predicted to an variable and we have shifted number of days to be predected in the dataset and keep that data empty so that we can predict that data, numpy array is used to convert the dataset into array format and we have assigned the opening price column to X variable and predicted price to Y variable, then split our hole data into 20% training data and 80% testing dataset, now we have two dataset that is training data and testing data, training data is used to train the machine learning model and testing dataset is used to test the machine learning model, after splitting the data we are fitting that dataset to our machine learning model that **Support Vector Machine(SVM) Regression ,** then accuracy score is found using the function (.score), now we are testing the trained model by passing the testing data set to the trained model and that model will give us the predicted data and that is compared with the testing data and we will plot the graph of testing data and machine learning model output data.

Modules used to implement this project:

1. Convolutional Neural Networks (CNNs): Effective for image classification tasks, including disease detection.

2. Transfer Learning: Utilize pre-trained CNN models (e.g., VGG16, ResNet50) and fine-tune for sugarcane leaf disease detection.

3. Support Vector Machines (SVMs): Suitable for small datasets and can handle high-dimensional data.

4. Random Forest: An ensemble learning method that combines multiple decision trees for accurate classification.

5. K-Nearest Neighbors (KNN): A simple, instance-based learning algorithm for classification tasks.

6. Deep Learning-based Models: Such as GoogleNet, InceptionV3, and DenseNet, which have shown promising results in image classification tasks.

7. Recurrent Neural Networks (RNNs): Can be used for detecting diseases from sequences of images or time-series data.

8. U-Net: A CNN architecture for image segmentation tasks, useful for detecting specific disease symptoms.

**Use case diagram**

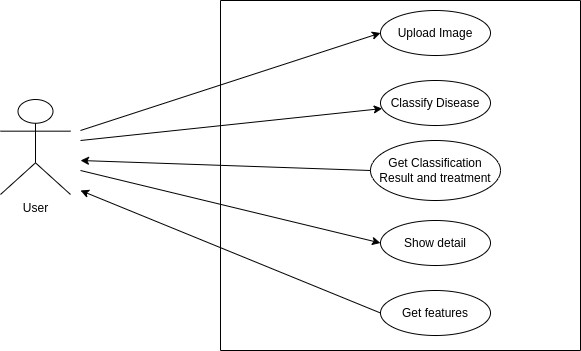


Figure 7.2: Architecture of Automated Sugarcane Diseases System

**CHAPTER-8**

**IMPLEMENTATION**

This chapter is about the background knowledge of our work, involving the basic concepts of deep learning and computer vision. Transfer learning 7.1 is used for pre-trained initialization in the experiment. Convolutional Neural Network (CNN) 7.2 Image Classification is applied to the Sugarcane Disease Dataset for our goals. Training Strategy 7.3 is the mathematical explanation of our experiment detail.

## Transfer Learning

Transfer learning is a powerful technique in deep learning that addresses the issues of model training and data dependency. With transfer learning, a model in a target domain can leverage pre-trained model parameters from another related task, serving as weight initialization for the target model. This process allows the model to transfer knowledge gained from one domain to another, reducing the need to train the target model from scratch.

After the pre-trained model parameters are initialized, the target model typically only needs to train its last layer or specific parts of the network to achieve convergence and improve performance. This fine-tuning step helps the model adapt to the specific requirements and tasks of the target domain. As a result, transfer learning significantly reduces training time and enhances the model's generality, making it applicable to various tasks.

In addition to mitigating the need for extensive training data, transfer learning also alleviates the dependency of deep learning models on large-scale data. Traditional deep learning models often require substantial amounts of data to understand underlying patterns. However, transfer learning enables the use of pre-trained models on related tasks, even if the training data and test data are not identically distributed. This capability makes transfer learning popular and widely used in deep learning, particularly in domains with limited training data availability.

Moreover, transfer learning is particularly beneficial when dealing with resource limitations. Re-implementing state-of-the-art models trained with massive data and multiple GPUs can be challenging for individuals with limited computing resources. In contrast, transfer learning allows for the utilization of pre-trained models, offering better performance with faster training speed.

In this work, transfer learning is applied as a pre-trained initialization for the models, utilizing pre-trained weight parameters from the ImageNet dataset. This approach is a crucial aspect of the controlled experiments conducted, addressing the concerns of training from scratch and data dependency.

* 1. **Pseudo code**

import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv)

# Input data files are available in the read-only "../input/" directory

# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

!pip install fastai

import fastai

import glob

from PIL import Image

import os

for dirname, \_, filenames in os.walk('/kaggle/input'):

for filename in filenames:

print(os.path.join(dirname, filename))

imgs=glob.glob("/kaggle/input/sugarcane-leaf-disease-dataset//.jpeg")

print(len(imgs))

from pathlib import Path

path=Path('/kaggle/working/dataset')

path.mkdir(exist\_ok=True,parents=True)

types=['Healthy','Mosaic','RedRot','Rust','Yellow']

for i in types:

dest=(path/i)

dest.mkdir(exist\_ok=True,parents=True)

def resize\_image(path,new\_size):

try:

with Image.open(path) as img:

resized\_img=img.resize(new\_size)

path=path.lstrip("/kaggle/input/sugarcane-leaf-disease-dataset/")

path=os.path.join('/kaggle/working/dataset',path)

resized\_img.save(path)

except Exception as E:

print(f"error:{E}")

print(imgs[1000])

new\_size=(300,200)

for i in imgs:

resize\_image(i,new\_size)

from fastcore.all import \*

from fastai.vision.all import \*

p="/kaggle/working/dataset"

dls=DataBlock(

blocks=(ImageBlock,CategoryBlock),

get\_items=get\_image\_files,

splitter=RandomSplitter(valid\_pct=0.2,seed=42),

get\_y=parent\_label,

item\_tfms=[Resize(192,method='squish')]

).dataloaders(p,bs=32)

dls.show\_batch(max\_n=12)

[10:24 AM, 7/27/2024] Vidya: learner=vision\_learner(dls,resnet18,metrics=error\_rate)

learner.fine\_tune(3)

from PIL import Image

is\_healthy,\_,probs=learner.predict(PILImage.create("/kaggle/input/sugarcane-leaf-disease-dataset/RedRot/redrot (10).jpeg"))

print(f"This is {is\_healthy}.")

learner.export('model.pkl')

**CHAPTER-9**

# TESTING

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that all the system elements have been properly integrated and perform allocated functions .The testing process is actually carried out to make sure that the product exactly does the same thing what is supposed to do. Testing is the final verification and validation activity within the organization itself. In the testing stage following goals are tried to achieve:-

* + To affirm the quality of the project.
  + To find and eliminate any residual errors from previous stages.
  + To validate the software as a solution to the original problem.
  + To provide operational reliability of the system.

During testing the major activities are concentrated on the examination and modification of the source code.

## Types of tests

* 1. **Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and its invasion. Unit tests perform basic tests at component level and test a specific business process, application and/or system configuration. Unit tests ensure that each unique path of a business process perform accurately to the documented specification and contains clearly defined inputs and expected results.

### Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing its event driven and it’s more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing. The combination of components is correct and consistent. Integration testing is specially aimed at exposing the problems that arise from the combination of components.

It is a testing in which the software under test is treated, as a black box, you cannot “see” into it. The test provides inputs and responds output without considering how the software works

### Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

### Test objectives

* All field entries must work properly
* Pages must be activated from the identified link
* The entry screen, messages and responses must not be delayed

### Features to be tested

* Verify that the entries are of the correct format
* No duplicate entries should be allowed All links should take the user to correct page.

## Functional testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

* + 1. Valid Input: identified classes of valid input must be accepted.
    2. Invalid Input: identified classes of invalid input must be rejected.
    3. Functions: identified functions much be exercised.
    4. Output: identified classes of application outputs must be exercised.
    5. Systems/procedures: interfacing systems or procedures must be invoked.

Organizations and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows, data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined

### System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process description and flows, emphasizing pre-driven process links and integration points.

### White Box Testing

White box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

### Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner working, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document.

## Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires Significant participation by the end user. It also ensures that the system meets the functional requirements. Testing the application from physically challenged person point of view as per (American Disability Act).

### Performance testing:

Performance testing for stock market price prediction involves evaluating the model's accuracy, robustness, and efficiency through metrics like MAE, MSE, and RMSE, using historical data for backtesting, walk-forward testing, and stress testing, while ensuring timely predictions and resource optimization in real-time conditions.

## Validation Testing

Validation testing for stock market price prediction using logistic regression involves several critical steps to assess the model's effectiveness and reliability. Firstly, the dataset is divided into training and testing sets, typically using a split such as 70% for training and 30% for testing. The logistic regression model is then trained on the training set, where features like historical prices, technical indicators, and sentiment scores are used to predict binary outcomes (e.g., stock price increase or decrease).

After training, the model's performance is evaluated using cross-validation techniques to ensure it generalizes well to unseen data. Cross-validation helps in understanding how the model performs across different subsets of the training data and provides insights into its stability and variance.

Summary

This code is well tested and all defected are removed from the code.

# CHAPTER-10

**SNAPCHATS**

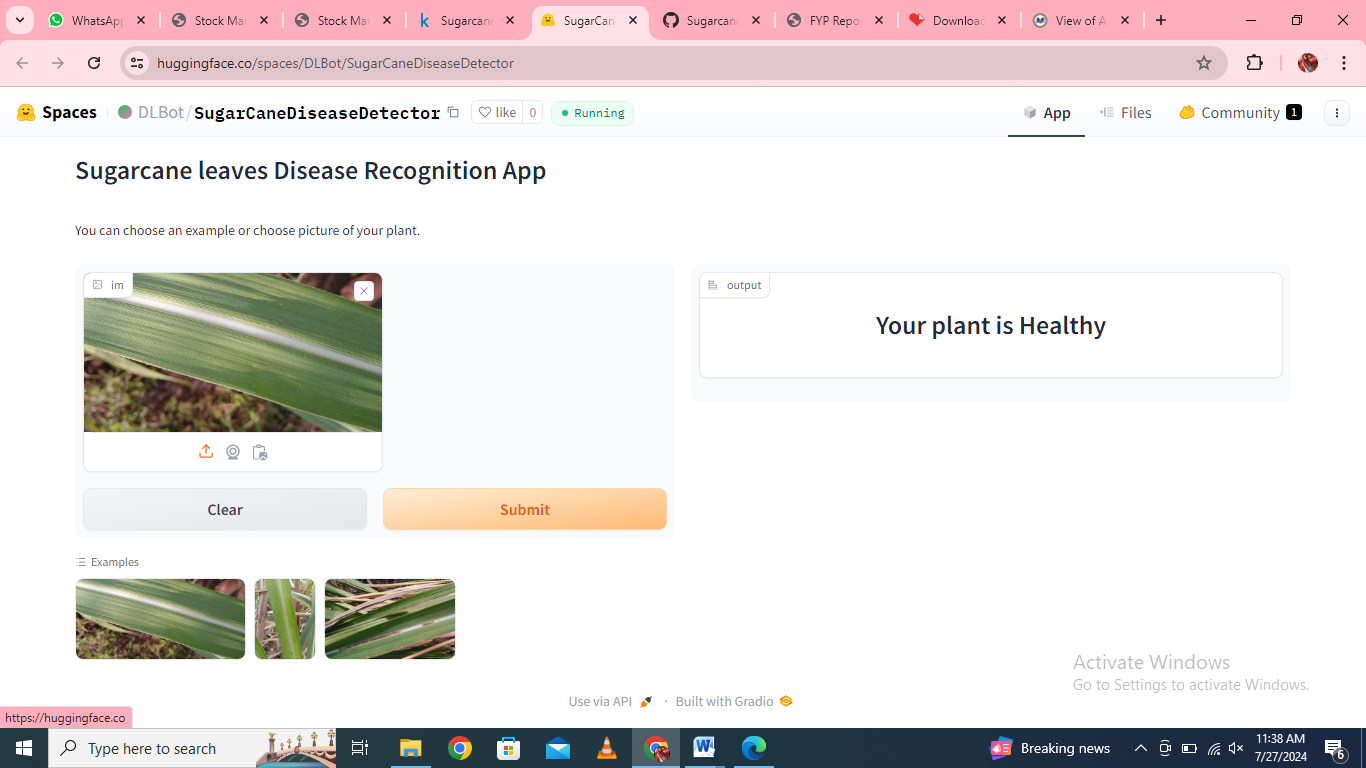


Fig 10.1 Healthy leaf

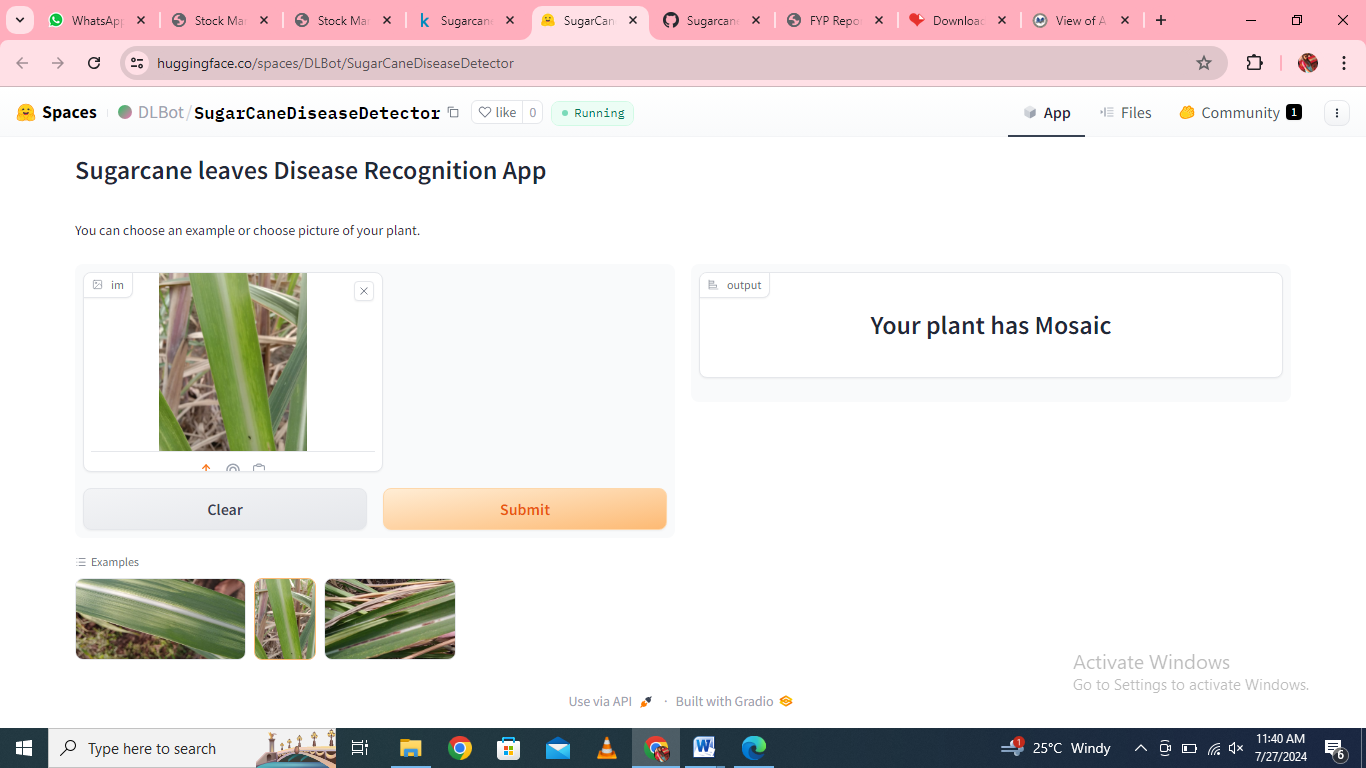


Fig 10.2 Mosaic disease

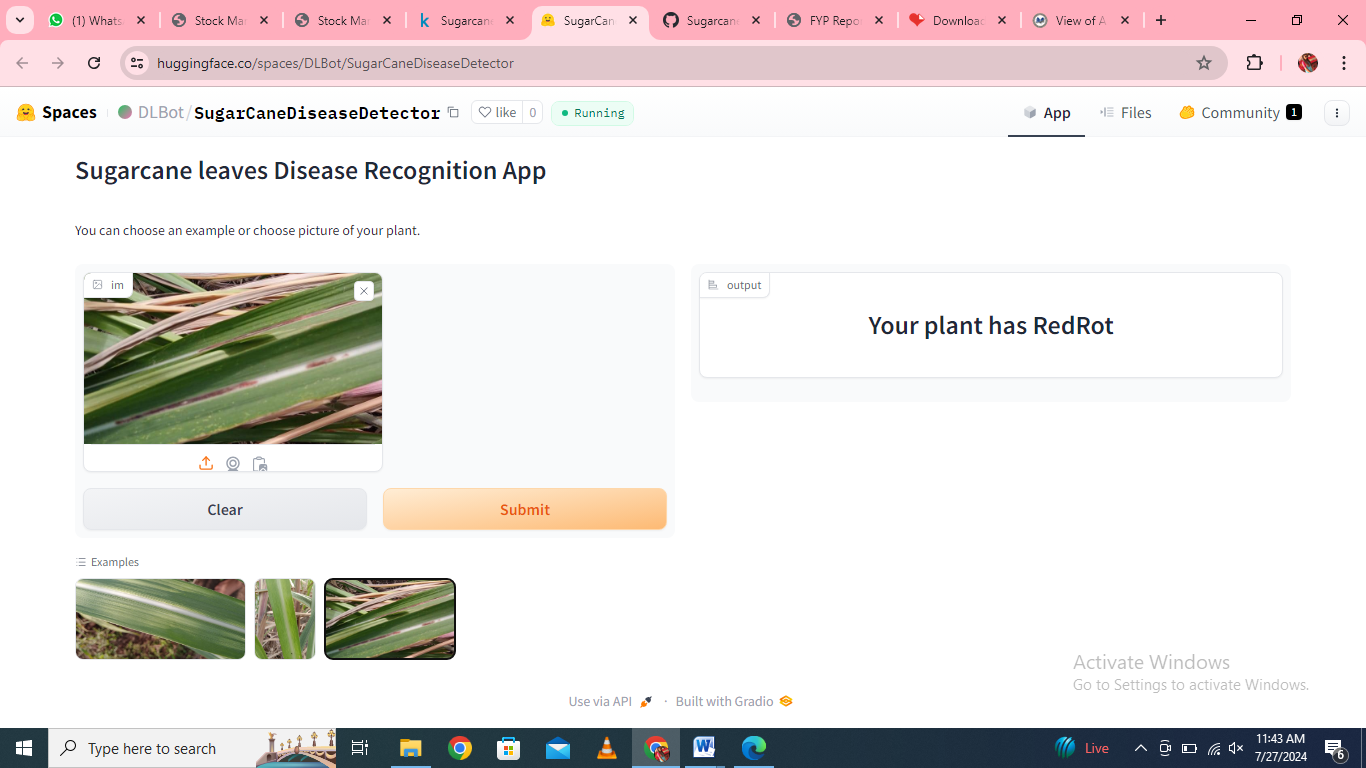


Fig 10.3 RedRot disease

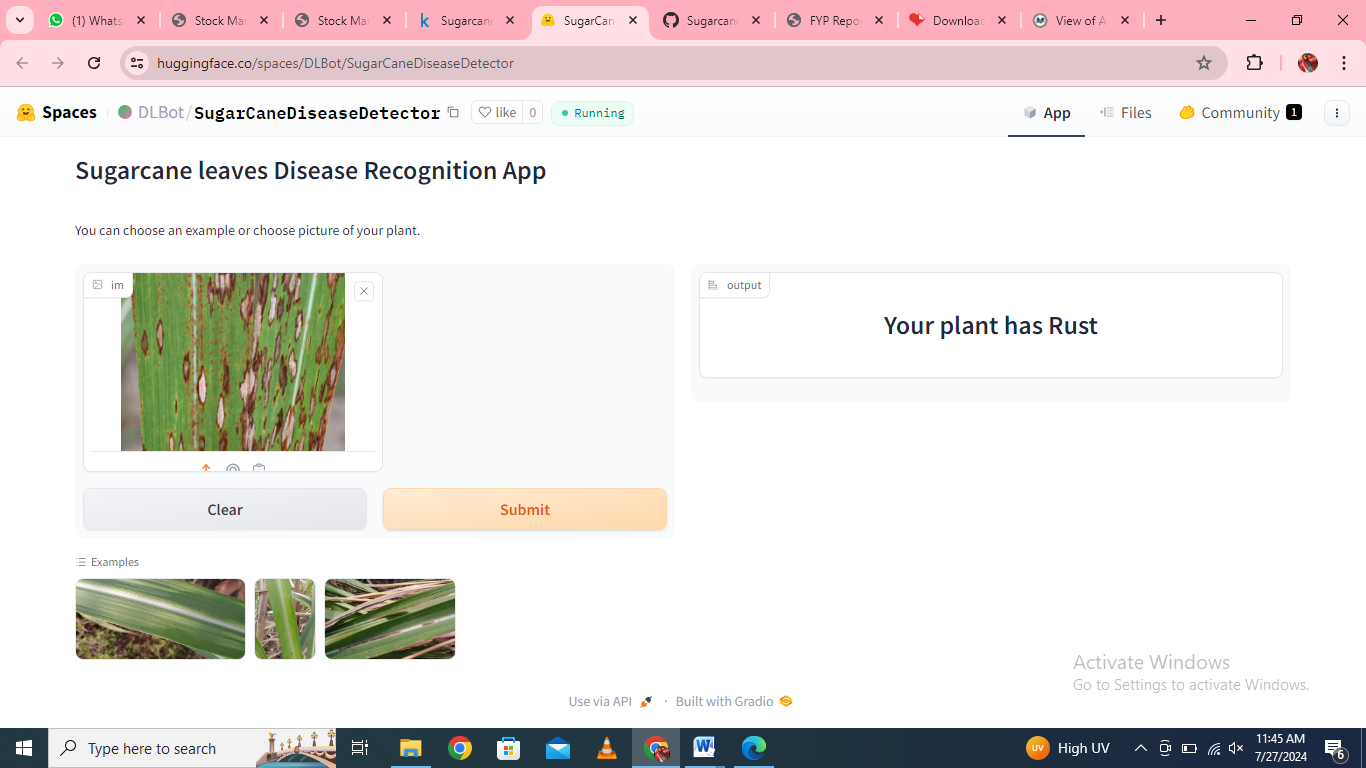


Fig 10.4 Rust disease

**CHAPTER-11**

**CONCLUSION**

The detection of sugar cane leaf diseases using Convolutional Neural Networks (CNNs) represents a significant advancement in agricultural technology, leveraging deep learning for improved accuracy and efficiency. Here's a summary of key points:

High Accuracy: CNNs can achieve high accuracy in identifying and classifying various sugar cane leaf diseases, thanks to their ability to learn intricate patterns from large datasets of labeled images.

Automated Detection: This approach allows for automated, real-time monitoring and diagnosis, reducing the reliance on manual inspections which can be time-consuming and prone to human error.

Early Detection: Early detection of diseases through CNNs can help in timely intervention, reducing crop loss and improving yield by allowing farmers to take preventive measures at an early stage.

Scalability: Once trained, CNN models can be deployed at scale, covering vast agricultural areas with minimal additional cost.

Adaptability: CNNs can be retrained and fine-tuned with new data, making them adaptable to emerging diseases and varying environmental conditions.

Cost-Effective: While the initial setup and training of CNN models may require investment, the long-term benefits include reduced labor costs and increased crop productivity.

Challenges: There are challenges such as the need for large annotated datasets, computational resources for training, and the model's ability to generalize across different conditions and regions.

In conclusion, CNN-based deep learning models offer a powerful tool for the detection of sugar cane leaf diseases, enhancing precision agriculture and potentially leading to significant improvements in crop management and productivity.

**REFERENCES**

1. Book: Field Guide – Diseases of Australian Sugarcane, Robert Magarey (2022)
2. Iqbal, Muhammad Aamir, and Asif Iqbal. "Sugarcane production, economics and industry in Pakistan." *Am. J. Agric.Environ. Sci*14.12 (2014): 1470-1477.
3. Hossain, Md Imam, et al. "Current and Prospective Strategies on Detecting and Managing Colletotrichumfalcatum Causing Red Rot of Sugarcane." *Agronomy* 10.9 (2020): 1253.
4. Kumar, Arpan, and Anamika Tiwari. "Detection of Sugarcane Disease and Classification using Image Processing." *Int J ResAppl Sci Eng Technol* 7.5 (2019): 2023-30.