GRAPE LEAF DISEASE IDENTIFICATION USING MACHINE LEARNING TECHNIQUES

Minor project-II report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

THUMATI HARSHITHA	(21UECS0631)	**(VTU20210)**
THUMATI VENGAL CHOWDARY	(21UECS0632)	**(VTU20212)**
NALLALA VINAY	(21UECS0404)	**(VTU20234)**

Under the guidance of
Mr. R. GANESAN, M.Tech.,
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2024

GRAPE LEAF DISEASE IDENTIFICATION USING MACHINE LEARNING TECHNIQUES

Minor project-II report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

THUMATI HARSHITHA	(21UECS0631)	**(VTU20210)**
THUMATI VENGAL CHOWDARY	(21UECS0632)	**(VTU20212)**
NALLALA VINAY	(21UECS0404)	**(VTU20234)**

Under the guidance of
Mr. R. GANESAN, M.Tech.,
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2024

CERTIFICATE

It is certified that the work contained in the project report titled "GRAPE LEAF DISEASE IDENTIFICATION USING MACHINE LEARNING TECHNIQUES" by "THUMATI HARSHITHA (21UECS0631), THUMATI VENGAL CHOWDARY (21UECS0632), NALLALA VINAY (21UECS0404)" has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor	**Signature of Professor In-charge**
Computer Science & Engineering	**Computer Science & Engineering**
School of Computing	**School of Computing**
Vel Tech Rangarajan Dr. Sagunthala R&D	**Vel Tech Rangarajan Dr. Sagunthala R&D**
Institute of Science and Technology	**Institute of Science and Technology**
May, 2024	**May, 2024**

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)
(THUMATI HARSHITHA)
Date: / /

(Signature)
(THUMATI VENGAL CHOWDARY)
Date: / /

(Signature)
(NALLALA VINAY)
Date: / /

APPROVAL SHEET

This project report entitled "GRAPE LEAF DISEASE IDENTIFICATION USING MACHINE LEARNING TECHNIQUES" by THUMATI HARSHITHA (21UECS0631), THUMATI VENGAL CHOWDARY (21UECS0632), NALLALA VINAY (21UECS0404) is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners **Supervisor**

Mr. R. Ganesan, M.Tech.,

Date: / /

Place:

ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO),D.Sc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice Chancellor Prof. S. SALIVAHANAN,** for providing us with an environment to complete our project successfully.

We record indebtedness to our **Professor & Dean, Department of Computer Science & Engineering, School of Computing, Dr. V. SRINIVASA RAO, M.Tech., Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

We are thankful to our respected **Head, Department of Computer Science & Engineering, Dr.M.S. MURALI DHAR, M.E., Ph.D.,** for providing immense support in all our endeavors.

We also take this opportunity to express a deep sense of gratitude to our **Internal Supervisor Mr. R. GANESAN, M.Tech.,** for his cordial support, valuable information and guidance, he helped us in completing this project through various stages.

A special thanks to our **Project Coordinators Mr. V. ASHOK KUMAR, M.Tech., Ms. U.HEMAVATHI, M.E., Ms. C. SHYAMALA KUMARI, M.E.,** for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

THUMATI HARSHITHA (21UECS0631)
THUMATI VENGAL CHOWDARY (21UECS0632)
NALLALA VINAY (21UECS0404)

ABSTRACT

Having diseases is quite natural in crops due to changing climatic and environmental conditions. Diseases affect the growth and produce of the crops and often difficult to control. To ensure good quality and high production, it is necessary to have accurate disease diagnosis and control actions to prevent them in time. Grape which is widely grown crop in India and it may be affected by different types of diseases on leaf, stem and fruit. Leaf diseases which are the early symptoms caused due to fungi, bacteria and virus. So, there is a need to have an automatic system that can be used to detect the type of diseases and to take appropriate actions. The proposed automatic system for detecting the diseases in the grape vines using image processing and machine learning technique. The system segments the leaf (Region of Interest) from the background image using Pre-processing. The Analyze and Predict layer. This layer will transform the inputs into a required standard format. Data preprocessing will consist of tasks such as feature selection, data cleaning, handle out of range values and missing values. Next layer is analyses and predict layer, this layer includes a trained model of for predicting the probability of occurrence of disease. The model is trained using a dataset of image samples of each disease. The output of this layer will be a probability of disease occurrence. The features are extracted from the segmented diseased part and it has been classified as healthy, rot, esca and leaf blight using different machine learning techniques such as Support Vector Machine (SVM), ada boost and Random Forest tree. Using SVM we have obtained a better testing accuracy of 93 percent.The results demonstrate the effectiveness of SVMs in grape leaf disease identification, offering a practical solution for disease management in vineyards. The achieved accuracy underscores the potential of machine learning techniques in precision viticulture, facilitating timely detection and intervention to mitigate crop losses.

Keywords:

Machine Learning, Image Processing,Support Vector Machine,Random Forest,Ada Boost

LIST OF FIGURES

LIST OF ACRONYMS AND ABBREVIATIONS

BR	Black Rot
GMM	Gaussian Mixture Model
GUI	Graphical User Inteface
HSV	Hue Saturation Value
IM	Image Processing
LB	Leaf Blight
ML	Machine Learning
PP	Preprocessing
RAM	Random Access Memory
RF	Random Forest
SVM	Support Vector Machine
UML	Unified Modelling Language

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Introduction

Grape is very commercial fruit of India. The grape plant will cause poor yield and growth. The diseases are due to the viral, bacteria and fungi infections which are caused by insects, rust. These diseases are judged by the farmers through their experience or with the help of experts through naked eye observation which is not accurate and time consuming process. Indian Economy is highly dependent on agricultural productivity of the country. Grape is very commercial fruit of India. It can easily be grown in all tropical, sub-tropical and temperate climatic regions. India has got different types of climate and soil in different parts of the country. This makes grapevines a major vegetative propagated crop with high socioeconomic importance. The grape plant will cause poor yield and growth when affected by diseases. The diseases are due to the viral, bacteria and fungi infections which are caused by insects, rust and nematodes etc., These diseases are judged by the farmers through their experience or with the help of experts through naked eye observation which is not accurate and time consuming process. Early detection of disease is then very much needed in the agriculture and horticulture field to increase the yield of the crops. The proposed system that can detect and identify diseases in the leaves of the grape plants.

1.2 Aim of the project

The aim of grape leaf disease identification using machine learning is to develop automated systems that can accurately detect and classify diseases affecting grape leaves. By leveraging machine learning algorithms, the goal is to create models capable of analyzing images of grape leaves and identifying signs of diseases such as powdery mildew, downy mildew, black rot, and others. These systems aim to provide early detection of diseases, allowing growers to intervene promptly and effectively to prevent the spread of infections and minimize crop losses. Furthermore, the development of such systems supports precision viticulture by enabling growers to monitor disease prevalence and distribution across vineyards with greater efficiency and accuracy. Ultimately, the aim is to reduce the dependency on manual inspection and provide grape growers with reliable tools for managing their crops more effectively, contributing to sustainable grape production and vineyard management practices.

In addition to accurate disease detection and classification, grape leaf disease identification using machine learning aims to enhance the efficiency and effectiveness of viticulture management practices. By automating the process of disease identification, growers can achieve several other important objectives. One aim is to enable proactive disease management by providing early warnings of disease outbreaks, allowing for timely intervention measures such as targeted pesticide application or pruning. This proactive approach helps minimize the spread of diseases, reduce crop losses, and optimize resource use, contributing to sustainable farming practices. Furthermore, by accurately identifying and categorizing different types of grape leaf diseases, machine learning systems can provide valuable insights into disease prevalence, distribution, and severity across vineyards. This information enables growers to implement tailored management strategies based on the specific needs of their vineyards, leading to more effective disease control and improved overall vineyard health.

1.3 Project Domain

Grape leaf diseases, such as powdery mildew, downy mildew, and black rot, pose significant threats to grapevine health and crop productivity. The core of the project involves implementing image recognition and classification algorithms, with a particular emphasis on support vector machine (svm) for effective image analysis. The dataset acquisition process involves overcoming challenges associated with collecting diverse and representative grape leaf images, followed by meticulous data labeling and annotation. The model training and validation phases are crucial, highlighting the significance of robust evaluation metrics. Beyond model development, the project explores the potential for real-time disease monitoring, contributing to early detection and proactive disease management.
Integration into existing agricultural practices is considered, emphasizing reduced

reliance on pesticides and improved yield predictions. The discussion extends to challenges such as environmental variability and the need for continuous model improvement, with ethical considerations and user interface development optionally addressed. Ultimately, the project aims to bridge the gap between cutting-edge machine learning and practical applications in agriculture, enhancing the efficiency and sustainability of grape cultivation.

1.4 Scope of the Project

The scope of the project is to mainly concentrate for the low cost and effectively diagnose the disease in grape leaves in faster and accurate ways.The grape leaf disease identification using machine learning encompasses the development of automated systems that can accurately detect and classify diseases affecting grape leaves. This involves a comprehensive exploration of image processing techniques to preprocess grape leaf images, including normalization, filtering, and segmentation. Feature extraction methods such as color histograms, texture analysis, and shape descriptors are employed to extract discriminative features from the images. Machine learning models, including support vector machines (SVMs), and decision trees, are trained and optimized to classify grape leaf diseases based on these features. The scope extends to dataset creation and augmentation to ensure comprehensive coverage of disease states and robust model training. Evaluation and validation of the models using appropriate metrics ensure reliable performance, and the ultimate aim is to deploy these models in practical settings for real-time disease monitoring and management in vineyards. Ongoing research and innovation drive advancements in methodology and technology to enhance the accuracy, efficiency, and scalability of grape leaf disease identification using machine learning.

Chapter 2

LITERATURE REVIEW

[1] Al-hiary et al., proposed this method for fast and accurate detection and classification of plant diseases (2021). An extension of this work will focus on developing hybrid algorithms such as genetic algorithms and NNs in order to increase the recognition rate of the final classification process underscoring the advantages of hybrid algorithms. By integrating the Z algorithm with a user-friendly interface, the authors facilitate the deployment of their system in real-world agricultural settings, enabling farmers and researchers to diagnose plant diseases effectively. The authors facilitate the deployment of their system in real-world agricultural settings, enabling farmers and researchers to diagnose plant diseases effectively.

[2] Guanlin et al., proposed that the ”Grape Leaf Disease Identification using Machine Learning Techniques,” (2019). An automatic leaf recognition system that identify diseases in grape leaves using machine learning technique.Presented at the international conference on computer and computing technologies in agriculture. Their research focuses specifically on the detection and classification of two prevalent grape diseases, downy mildew and powdery mildew, utilizing SVM algorithms for image recognition.he utilization of machine learning techniques, particularly SVM, in plant disease diagnosis has gained traction in recent years due to their ability to effectively classify complex and high-dimensional data.

[3] G.PremRishiKranth et al., proposed Plant disease prediction using machine learning algorithms. International Journal of Computer Applications (2022).Their study involves the collection of data on plant diseases, including symptoms, severity, and environmental parameters, which are then used to train and validate predictive models. By employing a range of machine learning algorithms, including decision trees, logistic regression, and k-nearest neighbors, the authors assess the accuracy and robustness of each model in predicting disease outbreaks across different crop species and geographical regions.

[4] H.Sabrol et al., proposed that the Tomato plant disease classification in digital images using classification tree. In: 2022 International Conference on Communication and Signal Processing (2022). In the study, five types of tomato diseases i.e. tomato late blight, Septoria spot, bacterial spot, bacterial canker, tomato leaf curl and healthy tomato plant leaf and stem images are classified.Their research focuses on employing classification tree algorithms for the automated identification of tomato plant diseases based on image analysis.The use of image processing and machine learning techniques for plant disease classification has gained considerable attention in recent years, driven by the need for efficient and non-destructive methods of disease diagnosis

[5] Nandini et al., proposed Pomegranate disease detection using image processing techniques (2021).Disease detection system for pomegranate leaves was proposed in which used colour-based segmentation and features like color, morphology and texture for classifying the leaves.Their study explores the utilization of WSMN to develop a web-based platform for detecting and monitoring plant diseases in agricultural settings.In recent years, there has been a growing interest in leveraging sensor networks and web-enabled interfaces to enhance crop monitoring, disease surveillance, and decision support systems in agriculture. Previous research has demonstrated the effectiveness of such systems in providing real-time data collection, analysis, and dissemination, enabling timely interventions to mitigate the impact of plant diseases on crop yield and quality.

[6] Sannakki et al., proposed Diagnosis and classification of grape leaf diseases using neural networks (2022). The detection and classification of leaf diseases in plants play a vital role in ensuring agricultural productivity and sustainability. Sannakki, S.S., Rajpurohit, V.S. Nargund, and V.B. Kulkan (2021) contribute to this field with their work on ”Detection and classification of leaf diseases using integrated approach of support vector machine and particle swarm optimization,”A system for two type of disease classification such as Downy mildew and Powdery mildew in grape leaves was proposed in using Back propogation Neural Network.Their study explores the utilization of a combined approach of SVM and Particle Swarm Optimization for the accurate identification and classification of leaf diseases.

[7] Suryawati et al., proposed Deep Structured Support Vector Machine for Tomato Diseases Detection. 2022 International Conference on Advanced Computer Science and Information Systems. Discuses about different disease classification techniques used for plant leaf disease and used genetic algorithm for image segmentation.Contributed to this field with their work on ”Deep Structured Support Vector Machine for Tomato Diseases Detection,” presented at the 2018 International Conference on Advanced Computer Science and Information Systems . Their study focuses on leveraging deep structured SVM for the accurate and automated detection of tomato diseases.

[8] Dr. Hiroshi Yamamoto, et al., (2023) examine studies leveraging machine learning models and image analysis algorithms to differentiate between disease symptoms and nutritional deficiencies, enhancing diagnostic accuracy and efficiency.The research landscape revealed a growing interest in leveraging advanced computational techniques to enhance diagnostic accuracy and efficiency in healthcare settings.Prior studies have highlighted the challenges associated with traditional diagnostic approaches, such as manual interpretation of medical images and subjective assessment of symptoms. These limitations underscore the need for automated solutions capable of extracting meaningful information from complex data sources.By harnessing SVM and other deep learning architectures, researchers have achieved remarkable success in accurately identifying disease markers and distinguishing them from benign conditions.Moreover, the integration of image analysis algorithms with deep learning frameworks has facilitated the development of robust diagnostic systems capable of handling diverse imaging modalities and clinical scenarios.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

The existing system proposed a segmentation method which has used mean-based strategy for computing threshold and textual features were extracted and classification was done by SVM. The survey proposed about the different disease classification techniques used for plant leaf disease and used genetic algorithm for image segmentation. An integrated approach of particle swarm optimization and SVM for plant leaf disease detection and classification was proposed in. Disease detection system for pomegranate leaves was proposed in which used color-based segmentation and features like color, morphology and texture for classifying the leaves.

Disadvantages of Existing System:

•By this methodology they had average accuracy in prediction.

•Used for plant leaf disease and used genetic algorithm for image segmentation

3.2 Proposed System

The proposed model is an automated disease detection and classification system for grape leaves using traditional image processing and machine learning techniques. The proposed system first segments the ROI from the back ground and classify the segmented leaves as healthy, black-rot, esca and leaf blight. Depicts different types of disease in grape leaves.Each disease has different characteristics where black rot appears to be circular in shape and has dark margins, esca appears as dark red stripes and leaf blight appears to be solid reddish-purple spots. The proposed system first captures the image converts the image into rectangular shape and detects whether it conists of any kind of disease if diseased part identified it results the name of the disease else results as healthy image.

Advantages of Proposed System:

• System consists of five different process such as image preprocessing, image segmentation, feature extraction, disease detection and identification.

• By using Bagging we will get better testing accuracy percentage.

3.3 Feasibility Study

3.3.1 Economic Feasibility

Cost/ benefits analysis of the project as over project is academic project we will not have only basic cost for learning of the technologies.

3.3.2 Technical Feasibility

Technical resources need for project Development

•ANACONDA

•TENSORFLOW

3.3.3 Social Feasibility

This assessment involves undertaking a study to analyze and determine whether and how well the organization's needs can be met by completing the project. Social feasibility studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development.

3.4 System Specification

3.4.1 Hardware Specification

Minimum hardware requirements are very dependent on the particular software being developed by a given Python / Canopy / VS Code user. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor.

• Operating system : windows11, linux

• Processor : minimum intel i3

- Ram : minimum 4 gb
- Hard disk : minimum 250gb

3.4.2 Software Specification

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation. The appropriation of requirements and implementation constraints gives the general overview of the project in regards to what the areas of strength and deficit are and how to tackle them.

- Anaconda 3.7

3.4.3 Standards and Policies

Anaconda Prompt

Anaconda prompt is a type of command line interface which explicitly deals with the ML(MachineLearning) modules.And navigator is available in all the Windows,Linux and MacOS.The anaconda prompt has many number of IDE's which make the coding easier. The UI can also be implemented in python.

Standard Used: ISO/IEC 27001

Chapter 4

METHODOLOGY

4.1 General Architecture For Grape Leaf Disease Identification

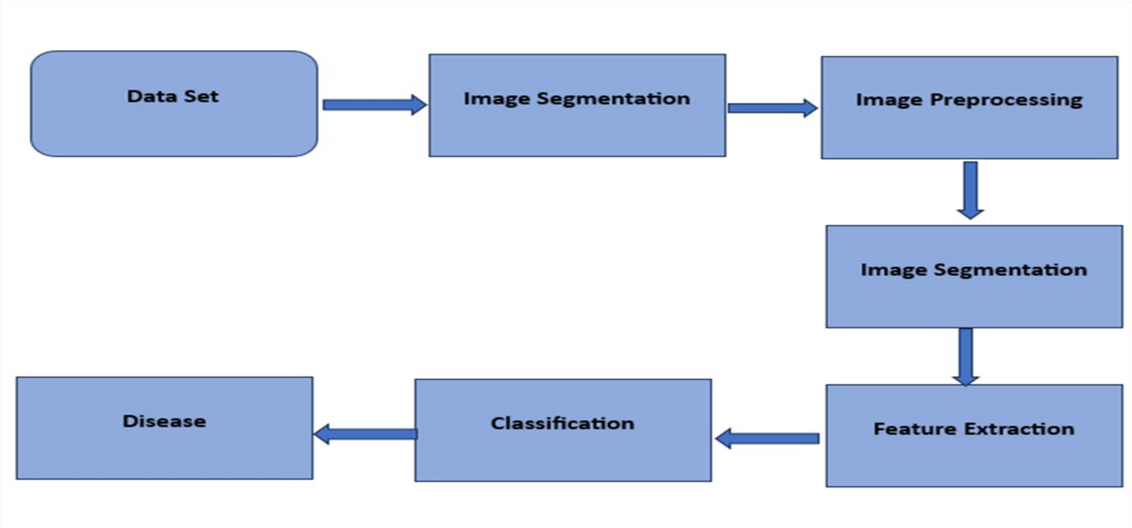

Figure 4.1: **Architecture Diagram of Grape Leaf Disease Identification**

The Fig 4.1 depicts the architecture diagram and gives the idea about the general architecture of grape leaf disease identification. First take the grape leaf from the dataset from here it goes to the image acquisition, then undergoes preprocessing.From segmented leaf part the disease region is furthur segmented based on two methods such as global thresholding and using semi-supervised technique.

4.2 Design Phase

4.2.1 Data Flow Diagram

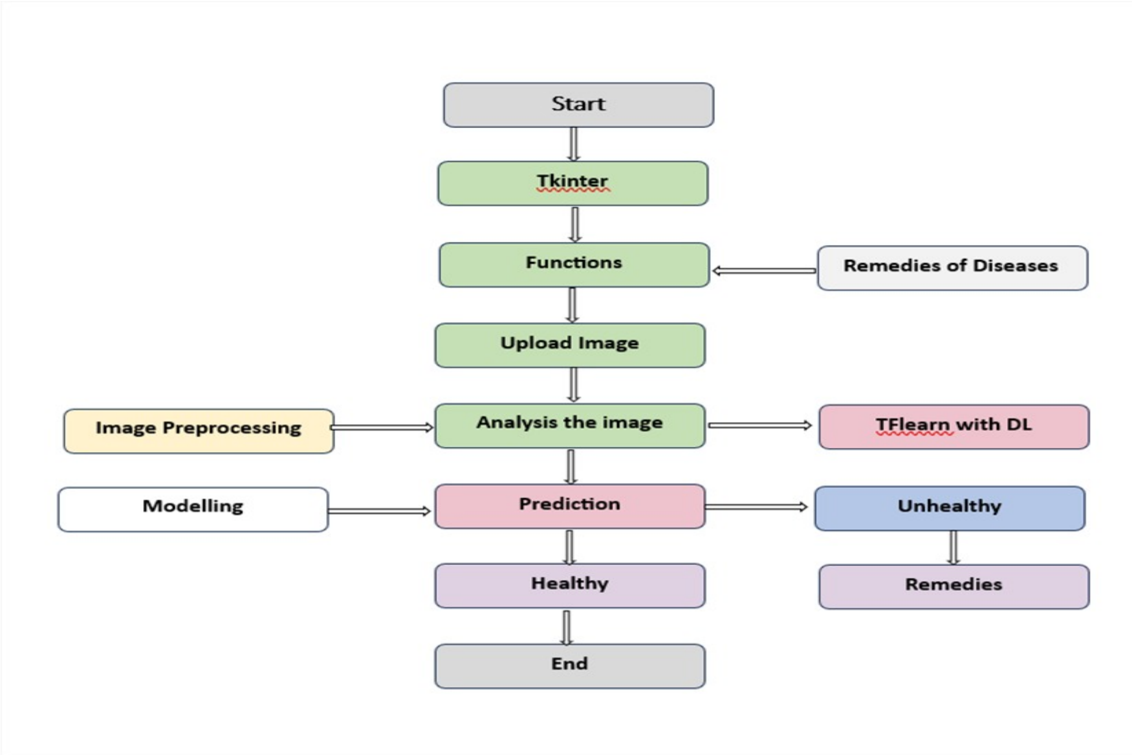

Figure 4.2: **Data Flow Diagram**

The Fig 4.2 depicts the data flow diagram, First we take a grape leaf and upload the image. Now we will analyse the image and predict the leaf is healthy or not healthy. If the leaf is healthy it will end there, if the leaf is unhealthy it will give the remedies.

4.2.2 Use Case Diagram

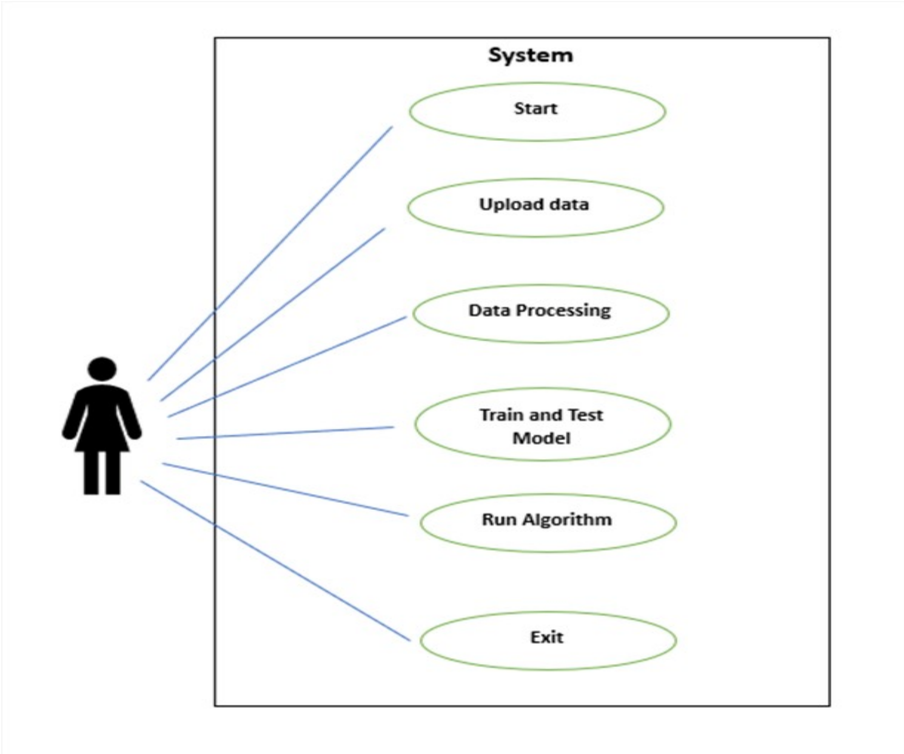

Figure 4.3: **Use Case Diagram**

The Fig 4.3 depicts the use case diagram that the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

4.2.3 Class Diagram

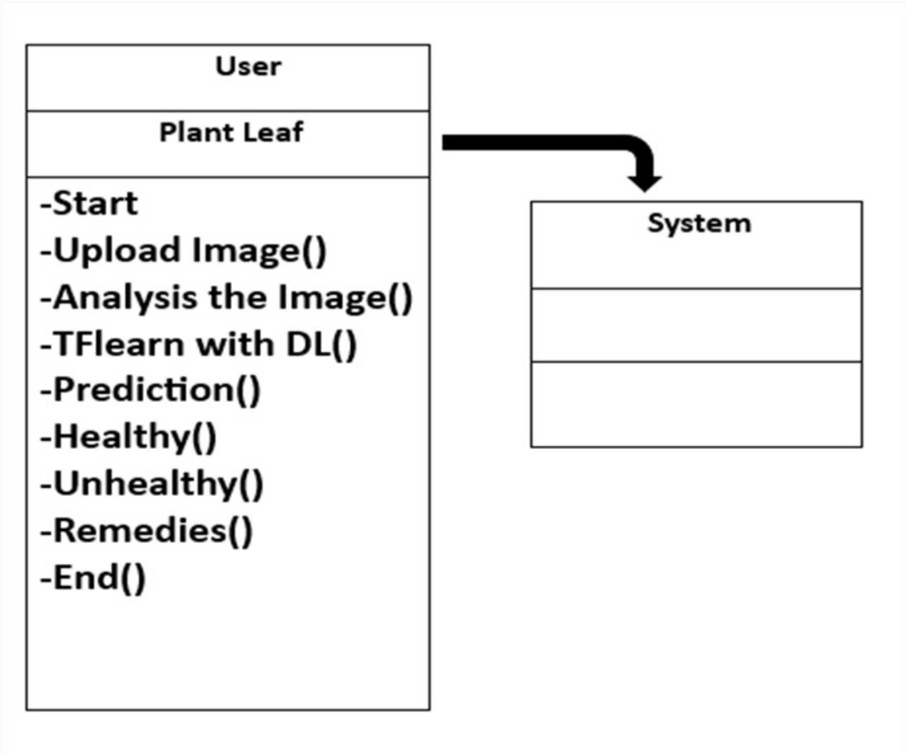

Figure 4.4: **Class Diagram**

The Fig 4.4 depicts the class diagram, describe that the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

4.2.4 Sequence Diagram

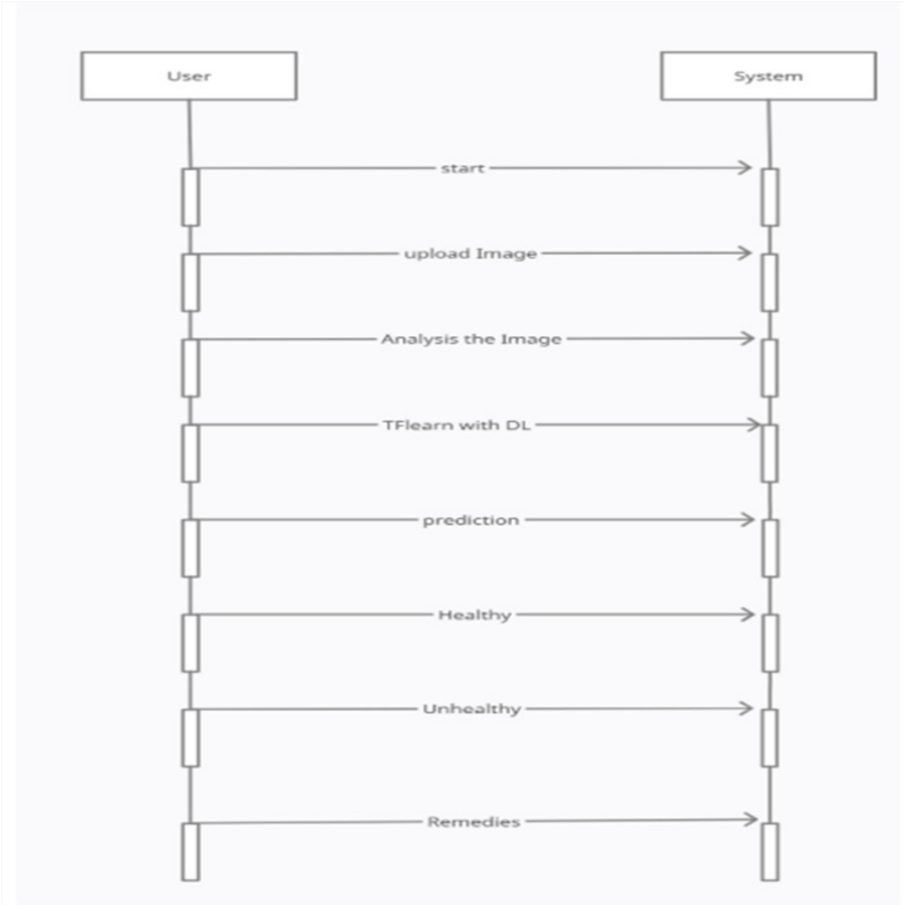

Figure 4.5: **Sequence Diagram**

The Fig 4.5 depicts the sequence diagram, describe that the Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart.

4.2.5 Activity Diagram

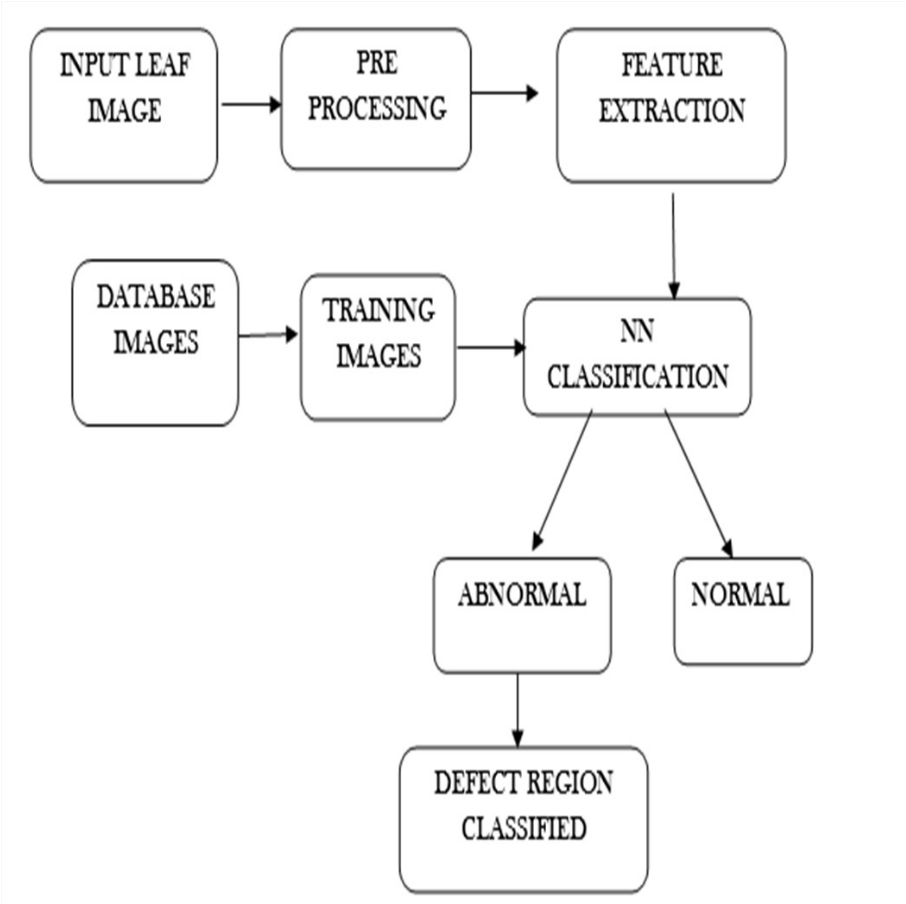

Figure 4.6: **Activity Diagram**

The Figure 4.6 depicts the activity diagram,the process begins with the acquisition of images of grape leaves from vineyards using cameras or other imaging devices. These images undergo preprocessing steps, including resizing, normalization, and noise reduction, to prepare them for analysis. After preprocessing, relevant features such as color, texture, and shape are extracted from the images. These features are then used to train machine learning models such as support vector machines (SVMs) in the model training phase. Once trained, the models are evaluated using validation datasets to assess their performance in terms of accuracy, precision, recall, and other metrics.

4.3 Enhanced SVM Algorithm

1. Define the Problem:
Identify the specific diseases you want to detect in grape leaves. Common diseases include leaf blight and black rot. Each disease will have unique visual symptoms on the leaves.

2.Collect and Prepare Data:
Gather a large set of images of grape leaves. These images should include healthy leaves as well as leaves affected by the diseases you are interested in.

3.Data Preprocessing:
Resize all images to the same dimensions since input data for a CNN needs to be of the same size. Normalize the pixel values (typically to a range of 0-1). Augment the data by applying transformations like rotation, zoom, and horizontal flipping to increase the robustness of the model.

4.Split the Dataset:
Divide your data into training, validation, and test sets:

Training set: Used to train the model.

Validation set: Used to tune the hyperparameters and avoid overfitting.

Test set: Used to evaluate the model performance independently.

5.Evaluate the Model:
After training the model, evaluate its performance using the test set. Calculate metrics such as accuracy, precision, recall to understand how well your model is performing in identifying different diseases.

6.Deploy the Model:
Once the model is well-trained and performs satisfactorily, it can be deployed in a practical application.

7.Monitor and Update:
After deployment, continue to monitor the model's performance and gather user feedback. Over time, you may need to retrain the model with new data.

4.4 Module Description

4.4.1 Preprocessing

The images are acquired from the web and are from different sources and sizes. The images also contains noise due to bad lightening condition, weather occlusion etc. To reduce the computational complexity the images are scaled down to a standard width and height.This scaled image are then processed to filter the noise using Gaussian filter. The Gaussian blur is a low pass filter that reduces the high frequency components, we have used 5*5 kernel size to filter the noise.

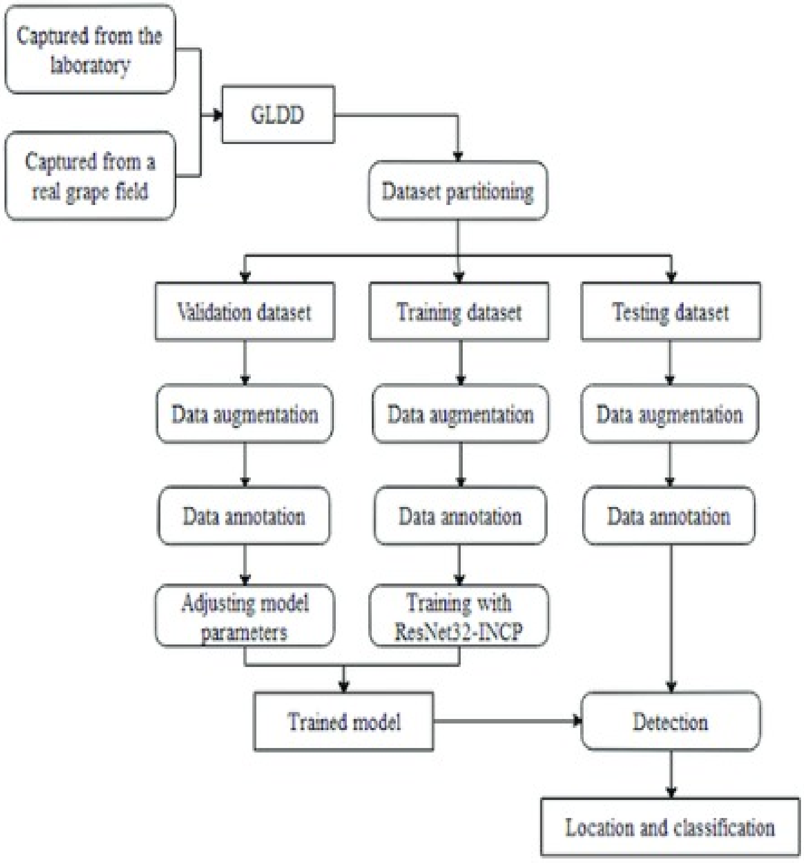

Figure 4.7: **Preprocessing**

4.4.2 Image Segmentation

From the preprocessed image, the leaf part of the image is segmented from the background image. This algorithm label a pixel as foreground or background using Gaussian Mixture Model (GMM) and also takes initial rectangle which is a rough segmentation between background and foreground. We have used a rectangle of dimension (10, 10, w-30 and h-20) as the bounding box where w and h are width and height of the image. From extracted foreground i.e. the leaf part, the diseased parts are extracted. The disease part contains lesions, colored spots and some yellowish part of the leaf. For extracting diseased region from the leaves we have proposed two different methods.

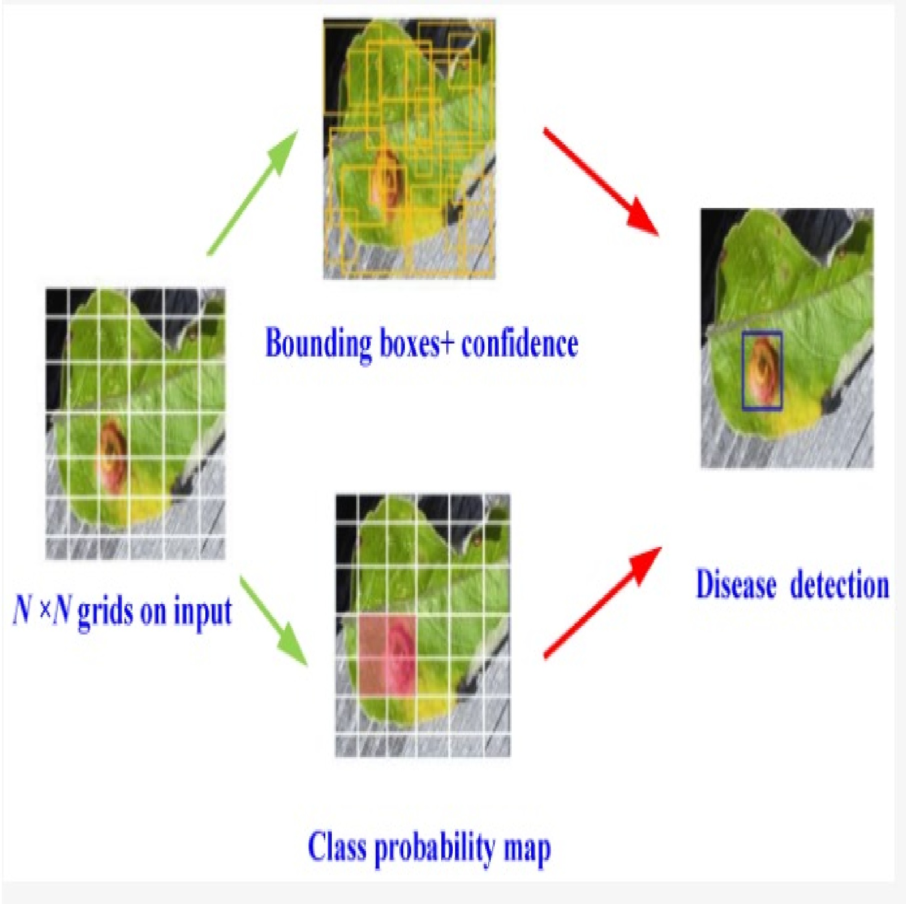

Figure 4.8: **Image Segmentation**

4.4.3 Diseased Part Identification

In this method, the RGB image is converted into grey scale image and then global thresholding is applied to convert the image into a binary image. On the threshold 13 image, connected component labeling is applied to find the contours. The contour with the largest area is then identified and morphological operations such as dilation and erosion is applied. The original image is converted to an black and white image and in the h channel, thresholding is applied. Binary AND operator is then applied to contour detected image . The resultant image was again threshold using binary invert thresholding.

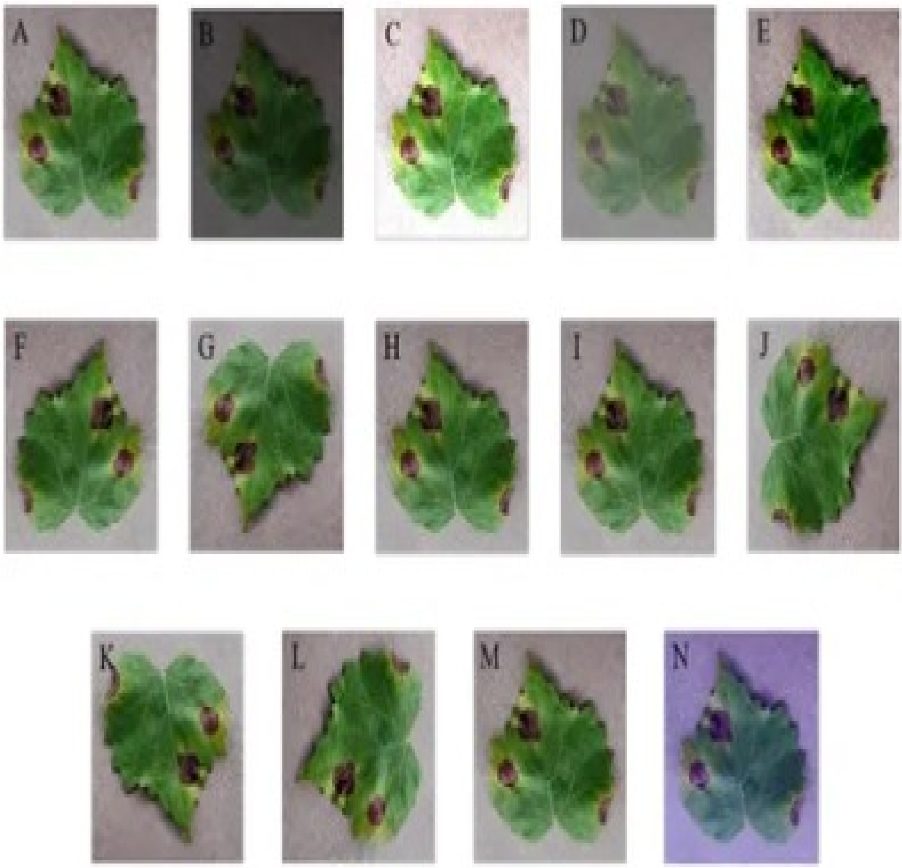

Figure 4.9: **Disease Part Identification**

4.4.4 Feature Extraction

Image features provide rich information about the content of the image. These features represent certain distinctive characteristics that can be used for differentiating among the categories of input patterns. In this work, we have used texture and color features of the images for classification. The texture of an image is usually expressed by contrast, uniformity, entropy etc. A statistical method of examining the texture of image is Gray-level Co-occurrence matrix (GLCM). GLCM extracts second order statistical texture features from the training images.

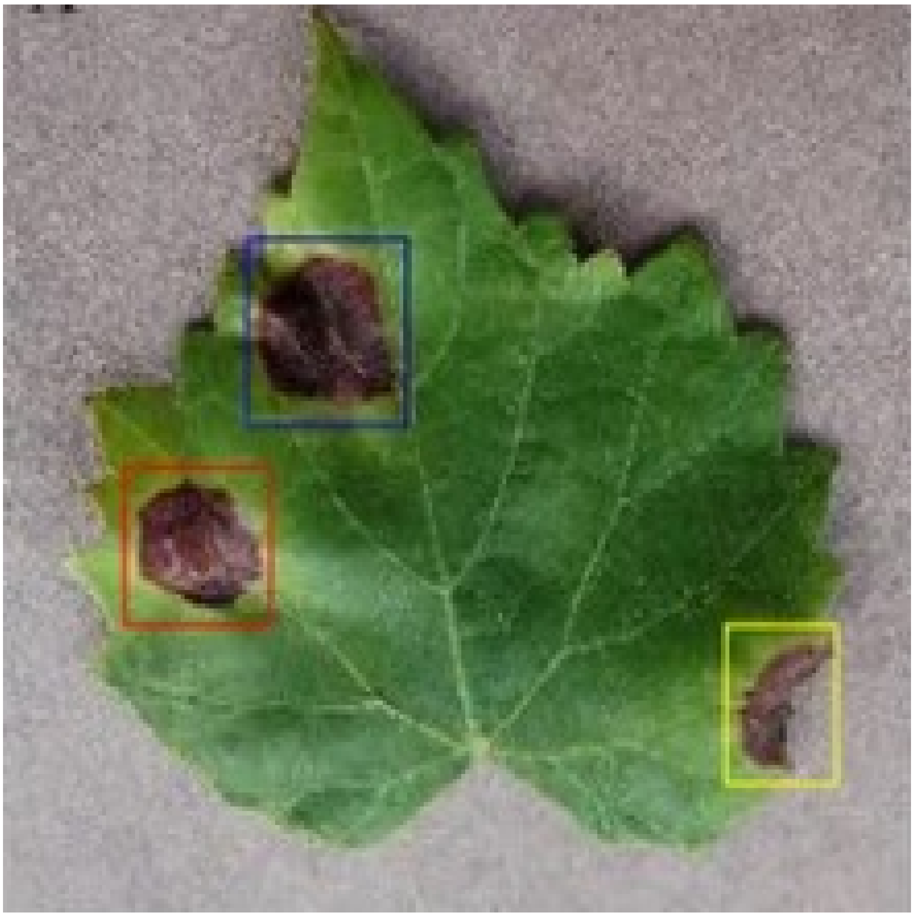

Figure 4.10: **Feature Extraction**

4.5 Steps to execute the project

Executing grape leaf disease identification using machine learning involves several key steps:

4.5.1 Gathering Dataset

Gather a diverse dataset of grape leaf images, including healthy and diseased samples.Clean and standardize the dataset, resizing images and applying data augmentation techniques.

4.5.2 Training the Model

Use a pre-trained model (e.g. SVM) to extract features from the images.Train the selected model using the training dataset. Fine-tune hyperparameters to achieve optimal performance. Use the validation set to monitor the model's performance and avoid overfitting

4.5.3 Evaluate and Testing the Model

Evaluate the model on a test set, fine-tune if necessary, and deploy for practical use. Monitor and update the model as needed for ongoing improvement.Document the entire process, including data collection, preprocessing steps, model architecture, and training details. This documentation is crucial for future reference and collaboration

Chapter 5

IMPLEMENTATION AND TESTING

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

5.1 Input and Output

5.1.1 Input Design

Input Design plays a vital role in the life cycle of software development, it requires very careful attention of developers. The input design is to feed data to the application as accurate as possible. So inputs are supposed to be designed effectively so that the errors occurring while feeding are minimized. According to Software Engineering Concepts, the input forms so screens are designed to provide to have a validation control over the input limit, range and other related validations.

5.1.2 Output Design

The output will be in the form such that in first module of the code the title Iot air pollution monitering system where readings can be monitered in screen where we can see pollution level gases where it present in atmosphere.Within fraction of seconds we can estimate the percentage of air quality index.

5.2 Testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to 15 check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

5.3 Types of Testing

5.3.1 Unit testing

In every stage of the project and fine-tuning the bug and module predicated additionally done by the developer only here.

Input

```
plt figure (figsize = (10, 10))
for image batchs , image labels in train ds take (1):
for i in range (9)
plt subplot(3,3, 1+1
plt axis('off')
plt imshow(image-batchs [i] numpy() astype ("uint8"))
plt title (class names[image labels [ilnumpy01)
```

5.3.2 Integration testing

The phase in software testing in which individual software modules are combined and tested as a group. It is usually conducted by testing teams.

5.3.3 System testing

It is type of software testing where by the system is tested against the functional requirements/specifications. Functions are tested by feeding them input and examining the output. Functional testing ensures that the requirements are properly satisfied by the application. This type of testing is not concerned with how processing occurs, but rather, with the results of processing. So, it tries to execute the test cases and compare the results and check the accuracy

5.3.4 Test Result

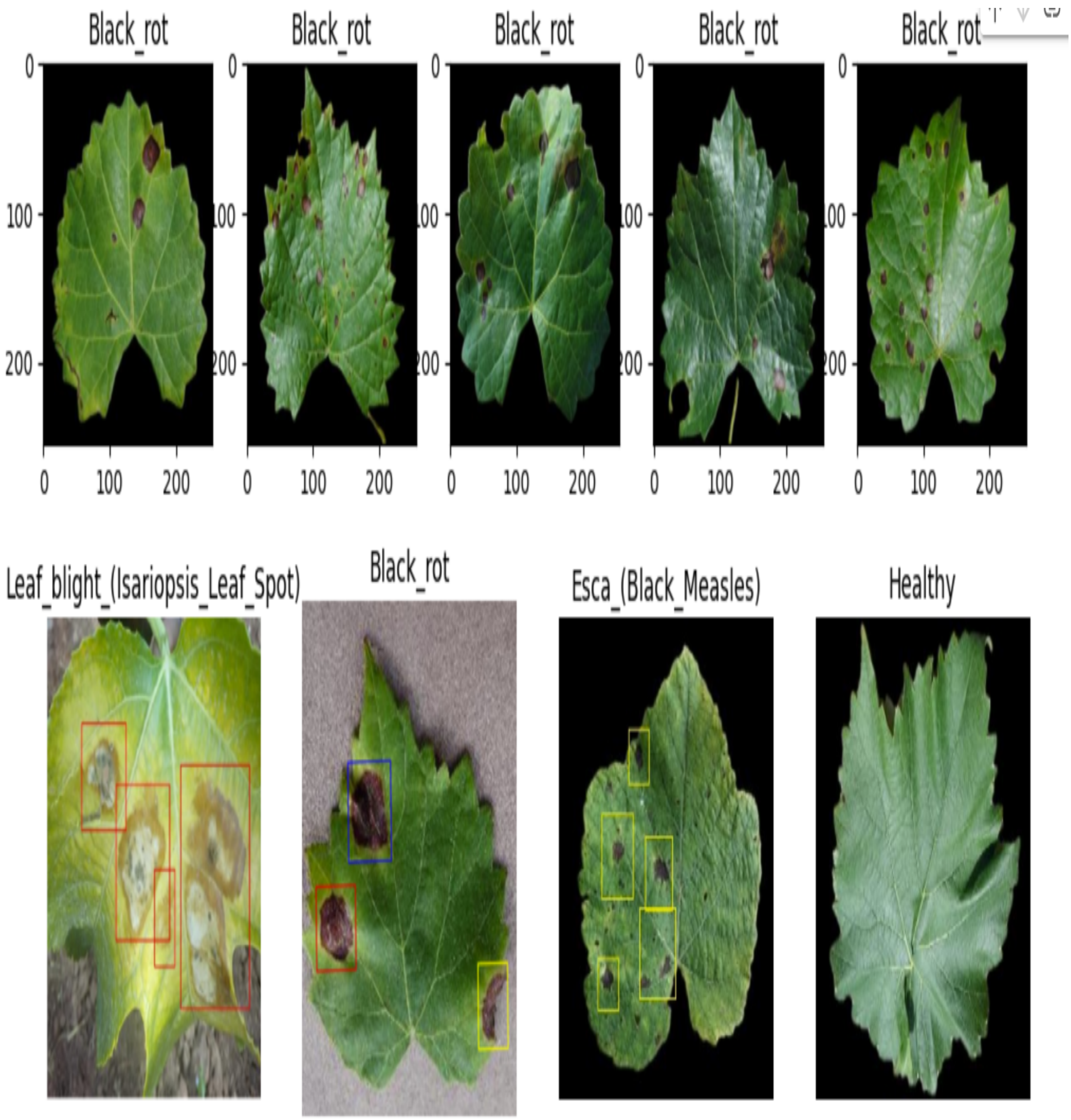

Figure 5.1: **Test Result of Grape Leaf**

The Fig 5.1 depicts the test result of grape leaf disease identification using Machine Learning

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

The proposed system first segments the leaf part from the background using Data preprocessing. From the segmented leaves diseased region are identified using two different methods. The first method is analyze layer whereas the second method is predict layer. From the identified diseased part texture and color features are extracted and trained using different classifiers and the results are compared. We have used SVM, Guassian model and image processing for classification. We have achieved a better result of 93.035as testing accuracy by using global thresholding and SVM.

6.2 Comparison of Existing and Proposed System

The Existing model is about an automated disease detection and classification system for grape leaves using traditional image processing and machine learning techniques. The proposed system first segments the ROI from the back ground using grab cut algorithm and classify the segmented leaves as healthy, black-rot, esca and leaf blight. Figure.1 depicts different types of disease in grape leaves. These diseases are caused due to fungi infection on the leaves. Each disease have different characteristics where black rot appears to be circular in shape and has dark margins, esca appears as dark red stripes and leaf blight appears to be solid reddish-purple spots.

6.3 Sample Code

```
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
dataset = tf.keras.preprocessing.image-dataset-from-directory('/content/drive/MyDrive/Grape leaf
    detection/train',
labels='inferred',
```

```
batch_size c = 32
image- c = (256, 256) ,
shuffle=True)
print(dataset)
len (dataset)
class names = dataset.class_names
class names
for image batchs, image labels in dataset.take (1):
print(image batchs. shape)
#
print(image-batchs.numpy())
print(image_labels.numpy())
def get dataset partitions tf (ds, ds size, train split=0.8, val split=0.1. test split=0.1, shuffle=
True, shuffle-size=10000):
assert (train-split + test-split + val-split) == 1
if shuffle:
# Specify seed to always have the same split distribution between runs
dsds. shuffle (shuffle size, seed=12)
train size = int(train split ds size)
val-size = int(val-split ds.size)
trai n\ ds=ds.take(train\ size)
valdsds.skip(train size). take (val size)
tes t\ ds=ds. skip (train\ size):skip(val\ size)
return train.ds, valds, test.ds
train ds, valds, test.ds = get_dataset partitions tf (dataset, 127)
print(tf.data.experimental.cardinality(train.ds). numpy())
print(tf.data.experimental.cardinality (val.ds). numpy())
print(tf.data.experimental.cardinality (test.ds).numpy())
plt.figure(figsize = (10, 10)
for image batchs. image labels in train ds.take (1):
for i in range (9):
plt.subplot(3.3,i+1)
plt.axis('off')
plt.imshow(image.batchs[i].numpy().astype("uint8"))
plt. title (class.names [image labels [i].numpy () ])
data.augmentation = tf.keras. Sequential ([
tf.keras.layers. RandomFlip(mode="horizontal and vertical", input shape=(256,256,3)).
tf.keras.layers. Random Rotation (0.1),
tf.keras.layers RandomZoom(.5.2)
1)
train.ds train.ds.cache().shuffle (1000). prefetch (buffer.size=tf.data.AUTOTUNE)
val.ds val.ds.cache().shuffle (1000). prefetch (buffer.size tf.data.AUTOTUNE)
test.dstest.ds.cache().shuffle (1000). prefetch (buffer_size tf.data. AUTOTUNE)
model tf.keras. models. Sequential ([
data augmentation.
tf.keras.layers. Rescaling (scale=1./255),
if. keras.layers. Conv2D(32, 3, activation="relu').
if.keras.layers. MaxPooling 2D ().
tf.keras.layers. Conv2D(64, 3. activation 'relu').
tf.keras.layers. MaxPooling 2D(),
```

```
tf.keras.layers. Conv2D(128, 3, activation -'relu"),
If.keras.layers. MaxPooling 2D ().
tf.keras.layers. Conv2D(256. 3. activation="relu").
tf.keras.layers. MaxPooling 2D ().
tf.keras.layers. Conv2D(512, 3, activation 'relu').
If.keras.layers. MaxPooling 2D (),
if. keras.layers. Flatten ().
tf.keras.layers. Dense (64, activation='relu").
if.keras.layers. Dense (32, activation='relu").
1)
tf.keras.layers. Dense (4, activation'softmax")
model. summary ()
model.compile(optimizer r = tf tf.keras.optimizers.Adam().
loss=tf.keras.losses. SparseCategoricalCrossentropy ().
metrics=['accuracy'])
run.logdir /runs/run_1/'
checkpoint.cb = tf.keras.callbacks. ModelCheckpoint(run_logdir+"CNN_1.h5", save_best_only=True)
tensorboard.cbif.keras.callbacks. TensorBoard (run.logdir)
history model. fit (
train_ds.
validation.data = val.ds.
epochs=1.
callbacks =[checkpoint_ch, tensorboard_cb]
dataset tf.keras.preprocessing.image-dataset-from-directory('/content/drive/MyDrive/Grape leaf = tf
    detection/test',
labels='inferred',
batch size=32,
image-size c = (256, 256)
shuffle=True)
plt.figure(figsize=(12, 12))
for images batch, labels batch in test.ds. take (1):
for i in range (9):
plt.subplot (3,3, i+1)
plt.imshow(images batch[i].numpy().astype('uint8'))
atch\ p re * diction =model.predict(images\ batch)
plt.title(f"Actual: {class names [labels batch[i]. numpy()]},\n Predicted: {class.names [np. argmax(
    batch_prediction [i])]}")
plt.axis("off")
plt.figure(figsize = (10, 10) )
for image batchs, image labels in train.ds.take (1):
for i in range (9):
plt.subplot(3,3, i + 1
plt.axis('off')
plt.imshow(image-batchs [i].numpy().astype ("uint8"))
plt.title(class names[image labels[ilnumpy01)
```

Output

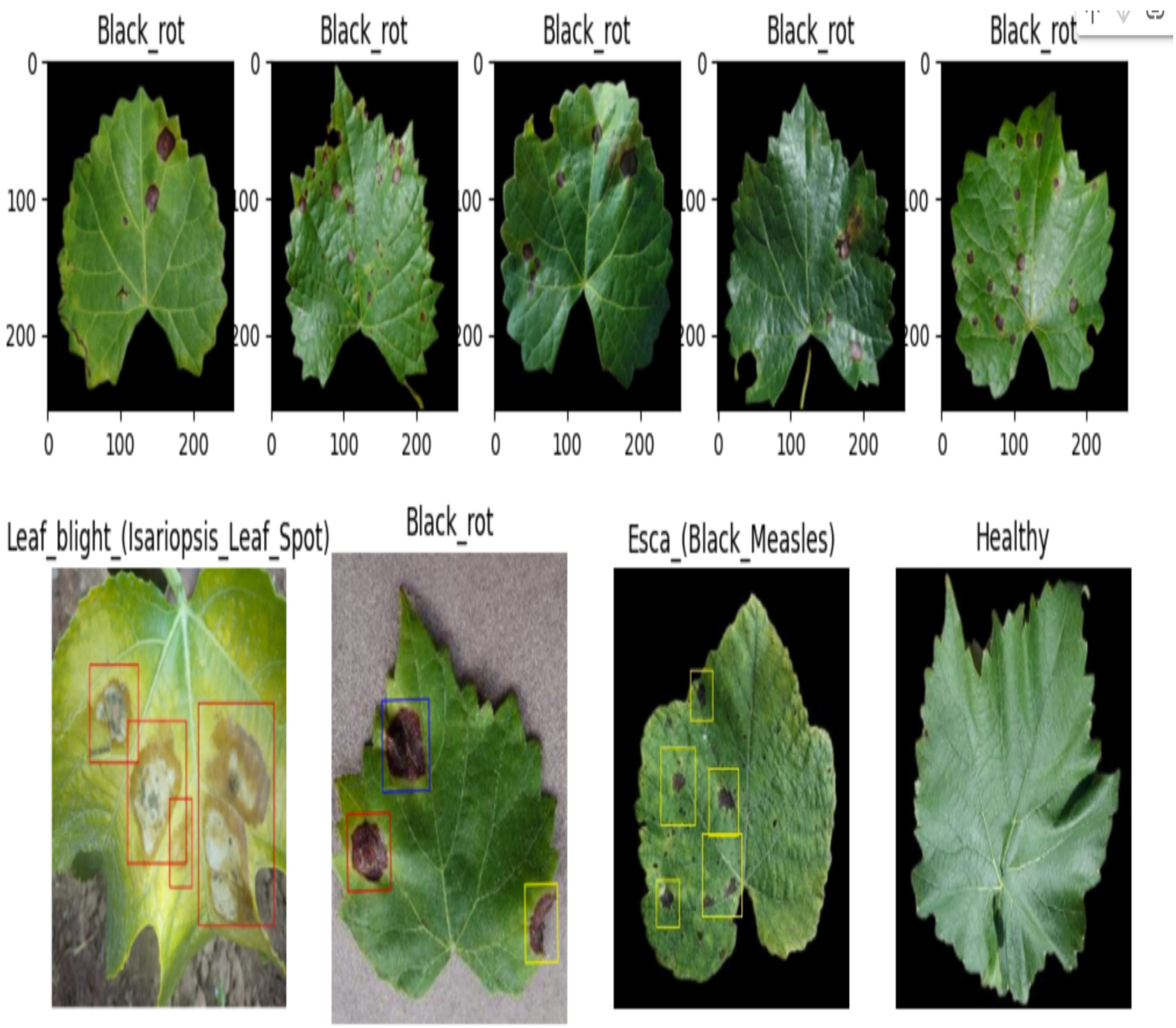

Figure 6.1: **Output for Grape Leaf**

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

The automatic leaf recognition system that identify diseases in grape leaves using machine learning technique. By harnessing the power of advanced algorithms and image processing techniques, researchers. This technological innovation offers several key advantages, including early detection of diseases, enabling timely intervention to mitigate yield losses. The proposed system first segments the leaf part from the background using grab cut segmentation technique.From the segmented leaves diseased region are identified using two different methods. The first method uses global thresholding technique whereas the second method using semisupervised learning technique. From the identified diseased part texture and color features are extracted and trained using different classifiers and the results are compared.The model SVM, random forest and Adaboost algorithms for classification. We have achieved a better result of 93.035 percent as testing accuracy by using global thresholding and SVM.

7.2 Future Enhancements

Future work can be developing the algorithm better segmented techniques, and by using more technologies of machine and learning. So that there is a scope of improvement in the techniques.This could be achieved through the collection of more diverse and detailed datasets, coupled with the development of sophisticated model architectures capable of capturing subtle variations in leaf symptoms. Moreover, the integration of multimodal data sources, such as spectral imagery and environmental data, could further enhance disease detection accuracy by providing a more comprehensive understanding of the factors influencing disease development.

Chapter 8

PLAGIARISM REPORT

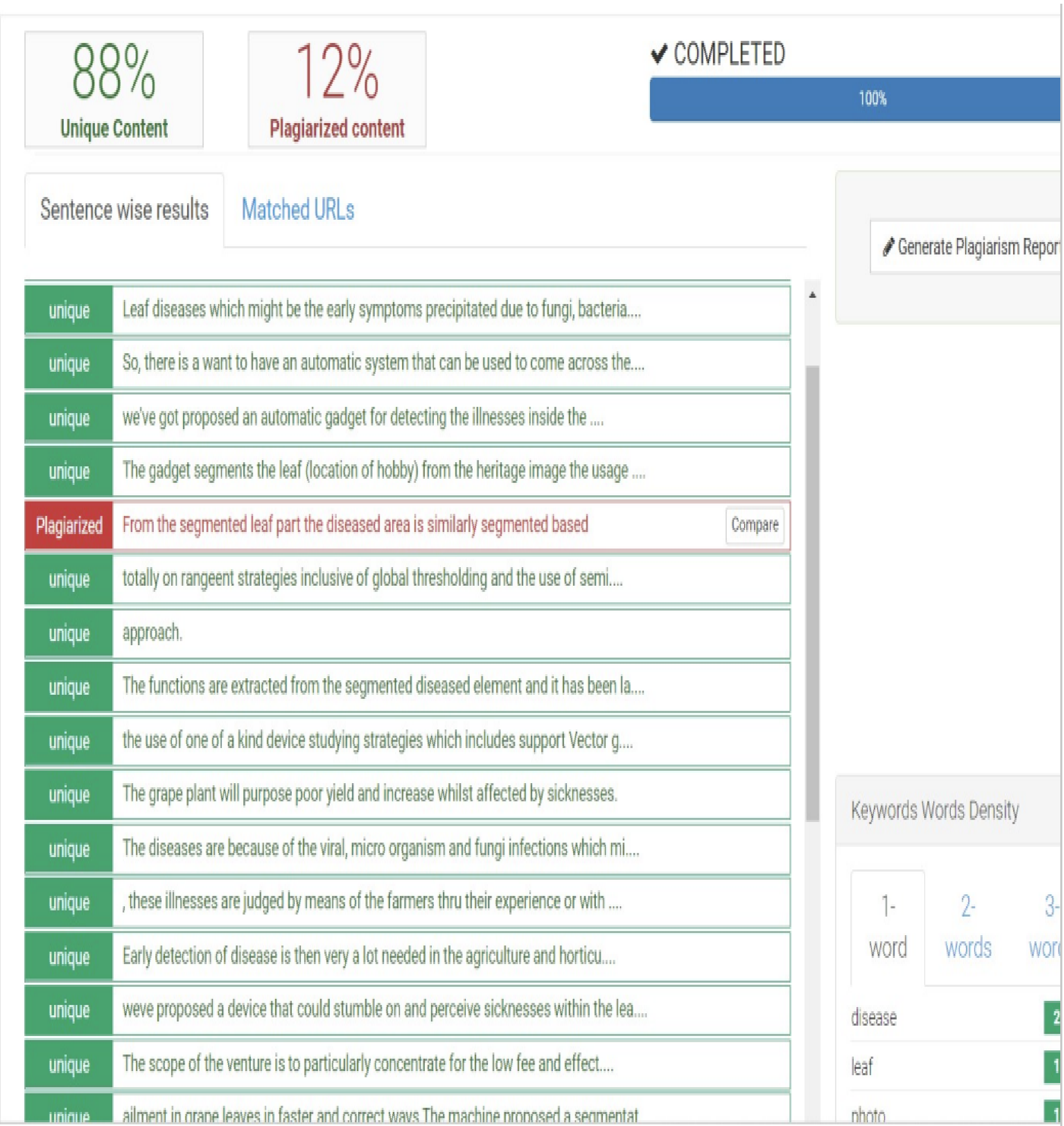

Figure 8.1: **Plagiarism**

Chapter 9

SOURCE CODE & POSTER PRESENTATION

9.1 Source Code

```
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
dataset = tf.keras.preprocessing.image-dataset-from-directory('/content/drive/MyDrive/Grape leaf
    detection/train',
labels='inferred',
batch_size c = 32
image- c = (256, 256) ,
shuffle=True)
print(dataset)
len (dataset)
class names = dataset.class_names
for image batchs, image labels in dataset.take (1):
print(image batchs. shape)
print(image-batchs.numpy())
def get dataset partitions tf (ds, ds size, train split=0.8, val split=0.1. test split=0.1, shuffle=
True, shuffle-size=10000):
assert (train-split + test-split + val-split) == 1
if shuffle:
dsds. shuffle (shuffle size, seed=12)
train size = int(train split ds size)
val-size = int(val-split ds.size)
valdsds.skip(train size). take (val size)
tes t\ ds=ds. skip (train\ size):skip(val\ size)
return train.ds, valds, test.ds
train ds, valds, test.ds = get_dataset partitions tf (dataset, 127)
print(tf.data.experimental.cardinality(train.ds). numpy())
print(tf.data.experimental.cardinality (test.ds).numpy())
plt.figure(figsize = (10, 10)
for image batchs. image labels in train ds.take (1):
for i in range (9):
plt.subplot(3.3,i+1)
plt.axis('off')
plt. title (class.names [image labels [i].numpy () ])
data.augmentation = tf.keras. Sequential ([
```

```
tf.keras.layers. RandomFlip(mode="horizontal and vertical", input shape=(256,256,3)).
train.ds train.ds.cache().shuffle (1000). prefetch (buffer.size=tf.data.AUTOTUNE)
val.ds val.ds.cache().shuffle (1000). prefetch (buffer.size tf.data.AUTOTUNE)
test.dstest.ds.cache().shuffle (1000). prefetch (buffer_size tf.data. AUTOTUNE)
model tf.keras. models. Sequential ([
data augmentation.
tf.keras.layers. Rescaling (scale=1./255),
if. keras.layers. Conv2D(32, 3, activation="relu').
if.keras.layers. MaxPooling 2D ().
tf.keras.layers. Conv2D(64, 3. activation 'relu').
tf.keras.layers. MaxPooling 2D(),
tf.keras.layers. Conv2D(128, 3, activation -'relu"),
If.keras.layers. MaxPooling 2D ().
tf.keras.layers. Conv2D(256. 3. activation="relu").
tf.keras.layers. MaxPooling 2D ().
tf.keras.layers. Conv2D(512, 3, activation 'relu').
If.keras.layers. MaxPooling 2D (),
tf.keras.layers. Dense (64, activation='relu").
if.keras.layers. Dense (32, activation='relu").
tf.keras.layers. Dense (4, activation'softmax")
model.compile(optimizer r = tf tf.keras.optimizers.Adam().
loss=tf.keras.losses. SparseCategoricalCrossentropy ().
metrics=['accuracy'])
checkpoint.cb = tf.keras.callbacks. ModelCheckpoint(run_logdir+"CNN_1.h5", save_best_only=True)
tensorboard.cbif.keras.callbacks. TensorBoard (run.logdir)
history model. fit (
train_ds.
validation.data = val.ds.
epochs=1.
callbacks =[checkpoint_ch, tensorboard_cb]
dataset tf.keras.preprocessing.image-dataset-from-directory('/content/drive/MyDrive/Grape leaf = tf
    detection/test',
labels='inferred',
batch size=32,
image-size c = (256, 256)
shuffle=True)
for images batch, labels batch in test.ds. take (1):
for i in range (9):
plt.imshow(images batch[i].numpy().astype('uint8'))
atch\ p re * diction =model.predict(images\ batch)
plt.title(f"Actual: {class names [labels batch[i]. numpy()]},\n Predicted: {class.names [np. argmax(
    batch_prediction [i])]}")
plt.axis("off")
for i in range (9):
plt.subplot(3,3, i + 1
plt.axis('off')
plt.imshow(image-batchs [i].numpy().astype ("uint8"))
plt.title(class names[image labels[ilnumpy01)
```

9.2 Poster Presentation

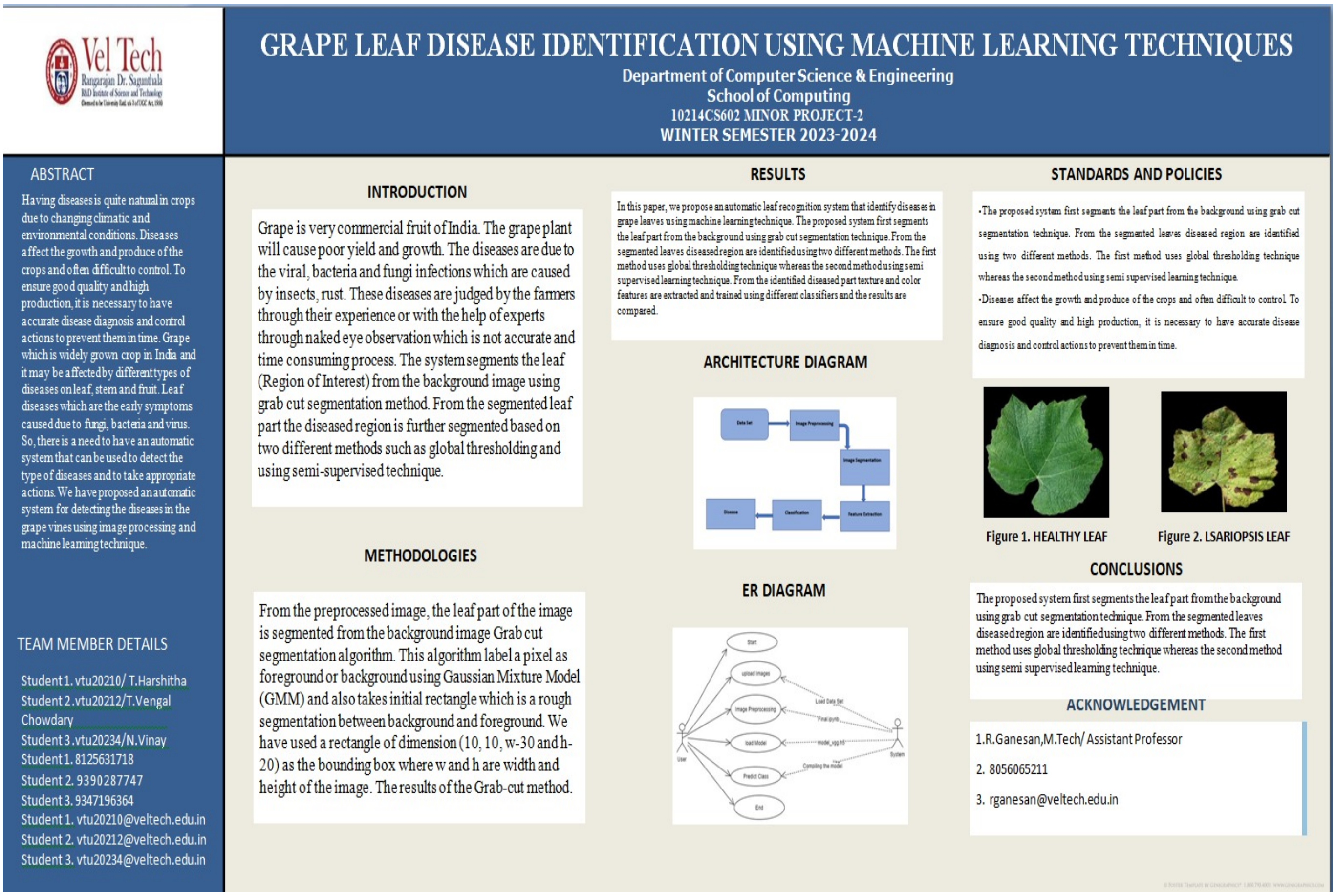

Figure 9.1: **Poster**

References

[1] Al-hiary, H., Bani-ahmad,S., Reyalat, M., Braik, M., Alrahamneh, Z.: Fast and accurate detection and classification of plant diseases. International Journal of Computer Applications 17(1) (2021).

[2] Guanlin, Zhanhong Ma, and Haiguang Wang," Image recognition of grape downy mildew and grape powdery mildew based on support vector machine." International Conference on Computer and Computing Technologies inAgriculture Springer, Berlin, Heidelberg, 2022.

[3] G.PremRishiKranth, HemaLalitha, LaharikaBasava, AnjaliMathurh: Plant disease prediction using machine learning algorithms. International Journal of Computer Applications 18(2) (2022).

[4] H.Sabrol, K.Satish: Tomato plant disease classification in digital images using classification tree. In: 2016 International Conference on Communication and Signal Processing (ICCSP), pp. 1242–1246 (2022).

[5] Nandhini, A., Hemalatha, Radha, Indumathi: Web enabled plant disease detection system for agricultural applications using wmsn. Wireless Personal Communications 102(2), 725–740 (2023).

[6] Sannakki, S.S., Rajpurohit, V.S., Nargund, V.B., Kulkarn: Detection and classification of leaf diseases using integrated approach of support vector machine and particle swarm optimization. 4(1), 79 – 83 (2021).

[7] Suryawati, E., Sustika, R., Yuwana, R. S., Subekti, A., Pardede, H. F. (2022). Deep Structured Convolutional Neural Network for Tomato Diseases Detection. 2018 International Conference on Advanced Computer Science and Information Systems (ICACSIS).