PROJECT REPORT

On

Image Based Plant Disease Detection

Mahavir Education Trust

Shah and Anchor Kutchhi Engineering College

Chembur, Mumbai

June-August 2021

ACKNOWLEDGEMENT

First and foremost, We would like to express our sincere gratitude towards the faculty of ‘Shah And Anchor Engineering College’, Mumbai without whose support and encouragement we would not have achieved what we have today. And a greatest thanks to our entire team evolved in this project.

We would like to extend our deepest thanks to our project guides, Mrs Dipti Mukadam and Mrs Bhakti Sonawane for providing us with their valuable support and contribution to this project. Their consistent support and Co - operation showed the way towards the successful completion of the project.

BIBLIOGRAPHY

[1]. Rauf, Hafiz Tayyab; Lali, Muhammad Ikram Ullah (2021), “A Guava Fruits and Leaves Dataset for Detection and Classification of Guava Diseases through Machine Learning”, Mendeley Data, V1, doi: 10.17632/s8x6jn5cvr.1

[2]. P. Revathi and M. Hemalatha, "Classification of cotton leaf spot diseases using image processing edge detection techniques," 2012 International Conference on Emerging Trends in Science, Engineering and Technology (INCOSET), 2012, pp. 169-173, doi: 10.1109/INCOSET.2012.6513900.

[3]. Mrs.Disha Sushant Wankhede, Dr. Selvarani Rangasamy, December 22, 2020, "Leaves: India’s Most Famous Basil Plant Leaves Quality Dataset", IEEE Dataport, doi: https://dx.doi.org/10.21227/a4f6-4413

[4]. M. Sardogan, A. Tuncer and Y. Ozen, "Plant Leaf Disease Detection and Classification Based on CNN with LVQ Algorithm," 2018 3rd International Conference on Computer Science and Engineering (UBMK), 2018, pp. 382-385, doi: 10.1109/UBMK.2018.8566635.

[5]. Shantanu Kumbhar, Amita Nilawar, Shruti Patil, Bodireddy Mahalakshmi, Manasi Nipane, “Farmer Buddy-Web Based Cotton Leaf Disease Detection Using CNN” International Journal of Applied Engineering Research ISSN 0973-4562 Volume 14, Number 11 (2019) pp. 2662-2666.

[6]. Tanmay A. Wagh, R. M. Samant, Sharvil V. Gujarathi, Snehal B. Gaikwad, “Grapes Leaf Disease Detection using Convolutional Neural Network” International Journal of Computer Applications (0975 – 8887) Volume 178 – No. 20, June 2019.

[7]. S.Yagneshwar Yadhav, T.Senthilkumar, S.Jayanthy, J.Judeson Antony Kovilpillai, “Plant Disease Detection and Classification using CNN Model with Optimized Activation Function” Proceedings of the International Conference on Electronics and Sustainable Communication Systems (ICESC 2020), IEEE Xplore Part Number: CFP20V66-ART; ISBN: 978-1-7281-4108-4.

[8]. G. Madhulatha, O. Ramadevi, “Recognition of Plant Diseases using Convolutional Neural Network” Proceedings of the Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC) IEEE Xplore Part Number:CFP20OSV-ART; ISBN: 978-1-7281-5464-0.

[9]. Keke Zhang, Qiufeng Wu, Yiping Chen, “Detecting soybean leaf disease from synthetic image using multi-feature fusion faster R-CNN”, Computers and Electronics in Agriculture 183 (2021) 106064.

[10]. Gavhale, Ms \& Gawande, Ujwalla. (2014). An Overview of the Research on Plant Leaves Disease detection using Image Processing Techniques. IOSR Journal of Computer Engineering. 16. 10-16. 10.9790/0661-16151016.

[11]. S. S. Chouhan, U. P. Singh, A. Kaul and S. Jain, "A Data Repository of Leaf Images: Practice towards Plant Conservation with Plant Pathology," *2019 4th International Conference on Information Systems and Computer Networks (ISCON)*, 2019, pp. 700-707, doi: 10.1109/ISCON47742.2019.9036158.

[12]. Vasumathi MT, Kamarasan M. (2021) An Effective Pomegranate Fruit Classification Based On CNN-LSTM Deep Learning Mod- els. Indian Journal of Science and Technology. 14(16): 1310-1319. https://doi.org/10.17485/IJST/v14i16.432.
[13]. P. Bansal, R. Kumar, and S. Kumar, “Disease Detection in Apple Leaves Using Deep Convolutional Neural Network,” Agriculture, vol. 11, no. 7, p. 617, Jun. 2021.
[14]. V. Feng, “An overview of resnet and its variants,” Medium, 17-Jul-2017. [Online]. Available: https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035. [Accessed: 24-Aug-2021].
[15]. “freecodecamp.org,” Certification | freeCodeCamp.org. [Online]. Available: https://www.freecodecamp.org/learn/machine-learning-with-python/#tensorflow. [Accessed: 24-Aug-2021].
[16]. M. Kawatra, S. Agarwal and R. Kapur, "Leaf Disease Detection using Neural Network Hybrid Models," 2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA), 2020, pp. 225-230, doi: 10.1109/ICCCA49541.2020.9250885.
[17]. M. Nikhitha, S. Roopa Sri and B. Uma Maheswari, "Fruit Recognition and Grade of Disease Detection using Inception V3 Model," 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), 2019, pp. 1040-1043, doi: 10.1109/ICECA.2019.8822095.
[18]. S. Kumar, K. Prasad, A. Srilekha, T. Suman, B. P. Rao and J. N. Vamshi Krishna, "Leaf Disease Detection and Classification based on Machine Learning," 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), 2020, pp. 361-365, doi: 10.1109/ICSTCEE49637.2020.9277379.
[19]. D. Tiwari, M. Ashish, N. Gangwar, A. Sharma, S. Patel and S. Bhardwaj, "Potato Leaf Diseases Detection Using Deep Learning," 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), 2020, pp. 461-466, doi: 10.1109/ICICCS48265.2020.9121067.

PREFACE

This report is an overview of a two month internship project we completed at **Shah and Anchor Kutchhi Engineering College, Mumbai**. The objective of this internship program was to familiarize us with implementation of knowledge we acquired in the curriculum. Since, the industry focuses mainly on practical skills, it is important for us to build our practical skills from our existing technical knowledge.

For this program, it was proposed to develop a project that would help detect different diseases in leaves. The project was developed keeping in mind that it will be usable and easily maintainable in upcoming years.

The report is made considering all the technical requirements and implementation details of the project. It also comprises the technical difficulties we faced during the development along with the future scope.

Developers

Project Synopsis
for
Image Based Plant Disease Detection

Name:	1. Omkar Dalvi 2. Rummaan Ahmad 3. Manish Gupta 4. Rahul Parande Engineering Students of Shah & Anchor Kutchhi Engineering College, Chembur, Mumbai
Project Name:	Image Based Plant Disease Detection
Duration:	Two Months (June 2021 to August 2021)
Name of Organisation: Address:	Shah and Anchor Kutchhi Engineering College, Mahavir Education Trust Chowk, W.T Patil Marg, D P Rd, next to Duke's Company, Chembur, Mumbai, Maharashtra-400088
Supervisor 1: Contact:	Prof. Bhakti Sonawane +91 9322307178
Supervisor 2: Contact:	Prof. Dipti Mukadam +91 9594435210
Project Type:	Image Processing (Python)

Introduction

Agriculture is one of India's most significant economic industries. It is the country's development's backbone. One of the most serious issues confronting the plant diseases are being discovered in the agriculture industry. In the past, disease detection was done by trained professionals. Farmers in distant locations have a tough time contacting specialists. Plant diseases are caused by a variety of factors, including climate change. There is a substantial loss in agricultural productivity in large farms if the illness is not identified at the correct time.

Machine Learning (ML) methods, Artificial Intelligence (AI), and Digital Image Processing (DIP) approaches have all exploded in popularity in recent years. It is critical for farmers to recognize various illnesses in their crops. During the early stages of development, it is necessary to adapt these new technologies to fit into today's world. Due to the proliferation of plant diseases, agricultural output will be lowered. These diseases can alter the form of the plant, cause harm to the plant, and

even kill it, demonstrating the influence on the colour and texture of the leaves, as well as the fruits, and so on. Some research has suggested several machine learning algorithms and visual processing approaches for illness detection and classification. Those diseases in plants are classified in several ways. For image processing, machine learning algorithms are created and pattern extraction, which can help with categorization and accuracy. This study uses the Convolutional Neural Networks (CNN) Algorithm to build an image processing method. To train, algorithms employ a variety of activation functions and then categorize the results.

Scope

The main aim of this project is to determine the diseased leaves along with their type. To implement this project Deep learning techniques such as Convolutional Neural Network (CNN) and Transfer learning are used. Apart from this a detailed analysis of different architectures of CNN along with their performances have also been investigated in an attempt to determine the optimal architecture for detecting disease in plant leaves.

Objectives

The objectives of the following project are:

- To have a program that can process images to detect diseases in plant leaves.
- To have a detailed comparative analysis of different architecture of CNN.
- To understand different concepts of Deep Learning and how to incorporate them to detect diseases in plant leaves.

Modules

The primary modules proposed are:

- Upload images.
- Classify images

Technology

The technologies used are:

- Python (Programming language).
- Google Collab and Pycharm (Code editors).

Image Based Plant Disease Detection

Software Requirements Specification

Table of Contents

1. INTRODUCTION

1.1 Project Background

Agriculture is one of India's most significant economic industries. It is the country's development's backbone. One of the most serious issues confronting the plant diseases are being discovered in the agriculture industry. In the past, disease detection was done by trained professionals. Farmers in distant locations have a tough time contacting specialists. Plant diseases are caused by a variety of factors, including climate change. There is a substantial loss in agricultural productivity in large farms if the illness is not identified at the correct time.

Machine Learning (ML) methods, Artificial Intelligence (AI), and Digital Image Processing (DIP) approaches have all exploded in popularity in recent years. It is critical for farmers to recognize various illnesses in their crops. During the early stages of development, it is necessary to adapt these new technologies to fit into today's world. Due to the proliferation of plant diseases, agricultural output will be lowered. These diseases can alter the form of the plant, cause harm to the plant, and even kill it, demonstrating the impact on the color and texture of the leaves, as well as its impact on the fruits, and so forth. Some research has suggested several machine learning algorithms and visual processing approaches for illness detection and classification. Those diseases in plants are classified in several ways. For image processing, machine learning algorithms are created and pattern extraction, which can help with categorization and accuracy. This study uses the Convolutional Neural Networks (CNN) Algorithm to build an image processing method. To train, algorithms employ a variety of activation functions and then categorize the results.

1.2 Project Scope

The main aim of this project is to determine the diseased leaves along with their type. To implement this project Deep learning techniques such as Convolutional Neural Network (CNN) and Transfer learning are used. Apart from this a detailed analysis of different architectures of CNN along with their performances have also been investigated in an attempt to determine the optimal architecture for detecting disease in plant leaves.

1.3 Purpose of this Document

This document explains all the functionality and environment of the GUI being developed, it serves the following purpose:

- Validation: It can be read by the user to check that the right product is being constructed.
- Verification: It is detailed enough to test whether the product built meets its specification.
- This document serves as a baseline for further design and development activity

1.4 Definitions/Glossary

- ML: Machine Learning
- AI: Artificial Intelligence
- DIP: Digital Image Processing
- CNN: Convolutional Neural Network

2. GENERAL DESCRIPTION

2.1 Context

It is proposed to help develop a software to detect the disease in leaves. This software will help detect disease in leaves with ease just by passing the picture of the leaf into the software by uploading the Image. The software will in turn classify the image as either healthy or diseased along with the type of disease.

The software would be available to run on the machine of the user and would be available to the general public.

2.2 Product Functions

FN01	Accepting the Image from the User to the Software	The Image taken by the user will be passed into the software through the upload button
FN02	Classifying the Leaf Image into its Categories	The Leaf Image uploaded will be classified by the Trained Model
FN03	Displaying the result of Classification	The result of classification by the model is passed to the User Interface and displayed

2.3 General and Design Constraints

The major design and implementation constraint has been the platform on which the software would execute. Due to the time constraint the software can only run on any machine having windows. But in the long run an application could be developed by using the trained model.

3. FUNCTIONAL REQUIREMENTS

3.1 FN01 Accepting the Image from User to the Software :

INPUT	Leaf Image uploaded by user
OUTPUT	The Image is displayed on Screen
PROCESSING	Pass the image to FN02

3.2 FN02 Classifying the Leaf Image into its Categories :

INPUT	Image of a Leaf is received for classification
OUTPUT	Classified result is generated
PROCESSING	Pass the result to FN03

3.3 FN03 Displaying the result of Classification :

INPUT	None
OUTPUT	The category of Leaf as per the Image provided
PROCESSING	The result after classification is displayed on the User Interface

4. DATA REQUIREMENTS

Since the model is trained on Guava Leaves taken from the dataset:

Rauf, Hafiz Tayyab; Lali, Muhammad Ikram Ullah (2021), “A Guava Fruits and Leaves Dataset for Detection and Classification of Guava Diseases through Machine Learning”, Mendeley Data, V1, doi: 10.17632/s8x6jn5cvr.1

Any image of a Guava leaf can be used to run the software for disease identification.

5. EXTERNAL INTERFACE REQUIREMENTS

5.1 User Interface

The GUI is Tkinter based where the user can upload an image and get the result based upon the classification.

1. Upload Button.
2. Classify Button.

5.2 Hardware Interfaces

- Windows PC (Minimum Requirement)
- Intel i3 or above / AMD Ryzen 3 or AMD FX8350
- 8GB DDR4 RAM
- 2GB VRAM

5.3 Software Interfaces

- Pycharm (v2019 and above),
- VS Code.

6. NON FUNCTIONAL REQUIREMENTS

6.1 Functional Requirements

- The application should be possible to users of Windows (7,8,10,11) and Mac (7,8,10,11).
- Since the GUI will be generating the results by analysing the image using the trained model , the response time for a particular analysis should be not greater than 3 - 4seconds.
- The update checks should not hinder the normal functioning of the GUI.
- Error handling should be implemented and the software should be able to handle all runtime errors.
- The software should be flexible for future enhancements, for example, the addition of new datasets, moving the interface to the web, etc.
- There should be synchronization with the latest data sources.

6.2 Software System Attributes

- User Friendly: The software to be designed should be user friendly. Operational level help should be provided in the software.
- Availability: The software should be available to the users until the specified expiry (if any) of the app.
- Maintainability: The software should be easy to understand and modify wherever necessary. For this sufficient comments should be provided to indicate the processing logic and other functionality.
- Portability: The software will be developed using Python under Windows operating system. The gui will be compatible for all Windows and Mac versions above the minimum specified version (7).
- Security: Security is maintainable through reaching following factors:
  a) To restrict unauthorized access.
  b) To keep control over S/W flaws.
  c) Data and resources should be protected.

Image Based Plant Disease Detection

Software Design Document

Table of Contents

Table Of Figures

1. INTRODUCTION

1.1 Project Background

Early identification of plant diseases is crucial for minimising losses in agricultural product output and quantity. The evaluation of visually visible patterns on the plant is required for the research of plant diseases. Plant disease diagnosis and surveillance are critical for long-term agriculture. It's quite difficult to keep track of plant diseases manually. It takes a great deal of effort, knowledge of plant diseases, and an inordinate length of processing time. As a result, image processing is being used to diagnose plant diseases. Image acquisition, image pre-processing, image segmentation, feature extraction, and classification are all processes in the disease detection process. It is easier and less expensive to identify illnesses automatically by just looking at the signs on the plant leaves. Machine vision is also supported, allowing for image-based automatic process control and inspection.

1.2 Project Scope

The main aim of this project is to determine the diseased leaves along with their type. To implement this project Deep learning techniques such as Convolutional Neural Network (CNN) and Transfer learning are used. Apart from this a detailed analysis of different architectures of CNN along with their performances have also been investigated in an attempt to determine the optimal architecture for detecting disease in plant leaves.

1.3 Purpose of this Document

This document explains all the functionality and environment of the GUI being developed, it serves the following purpose:

- Validation: It can be read by the user to check that the right product is being constructed.
- Verification: It is detailed enough to test whether the product built meets its specification.
- This document serves as a baseline for further design and development activity

1.4 References

The following references are considered to prepare this document.

Sr No	Title	Publisher/Author	Version	Release Date
1	SAMPARK SRS	NIC Mumbai	1.1	
2	SAMPARK SDD	NIC Mumbai	1.1	
3	SRS Template(NIC-TPL-002)	NIC Delhi	1.1	02/03/04
4	Project Life Cycle (NIC-PLC)	NIC Delhi	1.1	01/03/04
5	Quality Manual(NIC-QM)	NIC Delhi	1.0	29/02/04

1.5 Definitions/Glossary

- ML: Machine Learning
- AI: Artificial Intelligence
- DIP: Digital Image Processing
- CNN: Convolutional Neural Network

1.6 Structure of the Document

This document has 7 sections

Section 1	Instruction, project scope, purpose, document organization, references and definitions used in this document.
Section 2	General description of the design of proposed software.
Section 3	Data dictionary description.
Section 4	Program specifications.
Section 5	Security related information to be incorporated in the software.
Section 6	Operational control

2. GENERAL DESCRIPTION

2.1 Design Objectives

The design document of the Guava Leaf Disease Prediction has been created with the following objectives in mind:

- To find out Guava disease and help the farmers who grow Guava for their livelihood .
- To be a base for deriving a set of reusable design issues so that further evolutions can be carried forward.

2.2 Software Architecture

A detailed study of various architectures of CNN have been undertaken in order to generate the model which would not only serve the purpose but also give accurate results. The architectures trained are namely- ResNet50, Inception-V3, Inception-ResNet-V2 and VGG-19.

Characteristic	**ResNet50**	**Inception-V3**	**Inception-ResNet-V2**	**VGG-19**
Layers	50 (48, 1 MaxP, 1 AvgP)	48	164 (AvgP)	19 layers (16, 3, 5, 1)
Input Size	(224, 224, 3)	(224, 224, 3)	(224, 224, 3)	(224, 224, 3)
Accuracy (Epoch:10, BS:16)	81.82	93.18	90.91	88.64

Accuracy (Epoch:10, BS:32)	81.82	90.91	91.18	88.64
Accuracy (Epoch:11, BS: 16)	64.72	96.15	89.72	93.54
Accuracy (Epoch:11, BS:32)	84.09	93.18	92.45	90.91
Accuracy (Epoch:25, BS:32)	79.55	94.11	93.18	90.91
Output Nodes	5	5	5	5

The below graphs show the training accuracy and training loss of each Architecture -

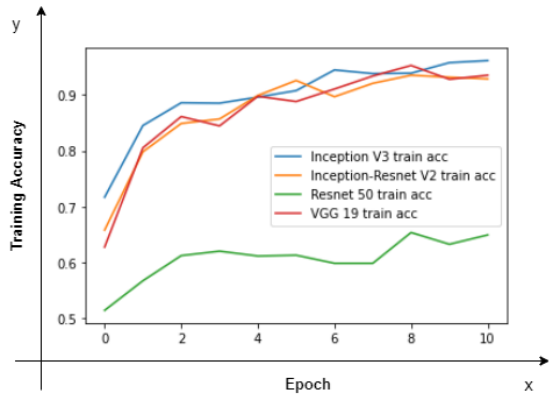

Figure 1: Training Accuracy Graph.

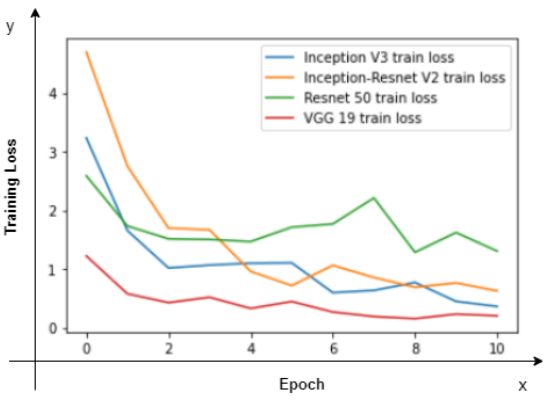

Figure 2: Training Loss Graph.

Based on a comparison of multiple CNN architectures, Inception V-3 had the best accuracy for our dataset, hence the application employs a pre-trained model to forecast the possible disease of the guava leaf. It takes an image as input and uses the dataset used to train the model to classify the type of disease and predict the same. The pre-trained model used is of Inception-V3 from the Tensorflow package and Keras sub package. Several network optimization strategies have been proposed in an Inception-V3 model to loosen the constraints and make model adaptation easier. Factorized convolutions, regularisation, dimension reduction, and parallelized computations are some of the techniques used. The architecture of the Inception-V3 model is built as:

1. Factorized Convolutions: This reduces the number of parameters in the Network and thus helps to reduce computations.
2. Smaller Convolutions: Convolutions of smaller size result in faster training of the model. For example a 5x5 filter consisting of 25 parameters is replaced to reduce training time by replacing it by 2 3x3 filters thus having only 18 parameters.
3. Asymmetrical Convolutions: Combinations of asymmetrical convolution layers outperform single symmetrical layers i.e they are faster.
4. Auxiliary classifier: A CNN layer is placed between layers during training as an auxiliary classifier, and the loss it incurs is added to the main network loss. In Inception-V3 an auxiliary classifier acts as a regularizer.
5. Grid Size Reduction: The grid size is reduced towards the end during Pooling operations. Before going through an average pooling layer the grid size is reduced by adding another filter to increase efficiency of the average pooling layer.

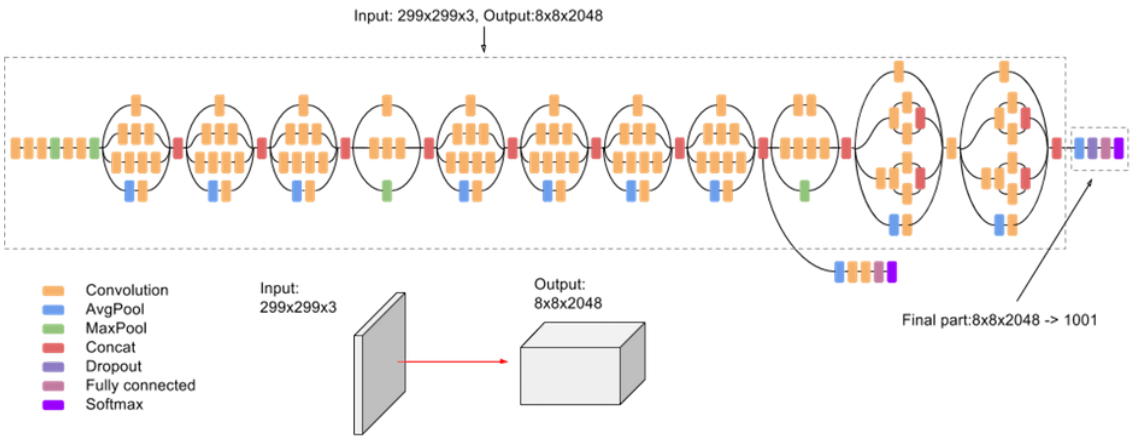

Figure 3: Inception V3 Architecture.
(Source: https://paperswithcode.com/method/inception-v3)

In this work, an Inception-V3 network is used for feature extraction as well as classification. Here the Inception-V3 is trained by the augmented dataset which contains 2337 images.

The batch size is set to 16 and 11 was the number of epochs. 60% of images from the augmented Guava leaves dataset were used to train the accuracy of this model. In every class , 20% of the images were selected for testing and 20% of the images were selected for validation. The testing dataset gives more than 95% accuracy. It means if 100 images are inputted then 95 images were classified correctly. The accuracy and the loss for both training and validation graphs generated by the model are shown below. When the training dataset is increased and epoch the accuracy is also increased. At the 9th epoch, the model gives the highest accuracy of 95.01%. Below figures show the training and validation accuracy and losses of the used model.

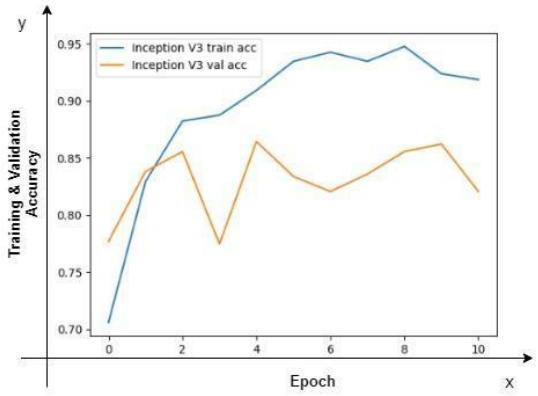

Figure 4: Inception V3 training and validation accuracy graph.

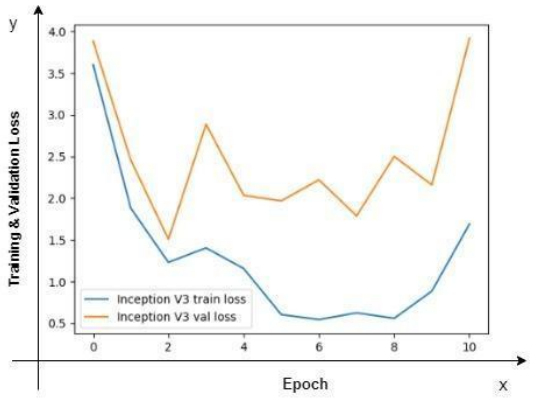

Figure 5: Inception V3 training and validation loss graph.

2.3 Operating Environment

The environment required for the operation of the system is as follows:

- **Hardware**: Laptop Or PC
- **Software:**
  Python runtime environment
  Python GUI – tkinter
  Tensorflow
  Keras

2.4 Design Constraints

The major design and implementation constraint has been the platform on which the software would execute. Due to the time constraint the software can only run on any machine having windows. But in the long run an application could be developed by using the trained model.

3. DATA DESCRIPTION

The Dataset required for training and testing the models have been taken from the following source:

Rauf, Hafiz Tayyab; Lali, Muhammad Ikram Ullah (2021), “A Guava Fruits and Leaves Dataset for Detection and Classification of Guava Diseases through Machine Learning”, Mendeley Data, V1, doi: 10.17632/s8x6jn5cvr.1

The dataset contained both healthy and diseased images of Guava fruit, leaves and stems. Diseased images were classified into 4 diseases namely - Canker, Rust, Mummification and Dot.

The total images of leaves from the above given source was 440 (After removing the fruit and stem images). Other details related to the images of the dataset are -

- File type : JPEG file
- Dimensions : 300 x 300 pixels
- Horizontal resolution : 96 dpi
- Vertical resolution : 96 dpi

Figure 6: Images in the Dataset.

Since the dataset is of size 440, it is not ideal enough to get accurate results. Data Augmentation technique was used to augment images of every category.

Images were augmented with a factor of 6 and total images after augmentation were found to be 2337 images.

The entire dataset is partitioned into training, testing and validation sets for both healthy and diseases in particular. The sets are partitioned in the ratio of 60% - 20% - 20% (.6.2.2) using the split function (present in the split function folder in the mentioned Github links).

Figure 7: Images in the Augmented Dataset

In order to access the augmented dataset kindly follow the below drive link:

https://drive.google.com/drive/folders/1ED-DaaP9tOOtG7jIUWJ3i35yYZXwlZFa?usp=sharing

Below table shows the quantity of Images in the Dataset:

Type	Quantity (Before Augmentation)	Quantity (After Augmentation)
Healthy	277	1289
Canker	42	278
Dot	36	216
Mummification	52	275
Rust	33	279

4. Program Specification

PPYTHON#1	split.py
Objective	To split the given data set into training, testing and validation in the specified ratio (.6.2.2)
Calling Function	None
Folders accessed	dataset1

PPYTHON#2	model1.py
Objective	To train the model using transfer learning
Calling File	gui.py
Folders accessed	dataset2

PPYTHON#3	gui.py
Objective	To take the image as input and display the result as output
Calling Function	None
Model accessed	model_inception.h5

PJPTERNB#1	Comparative analysis.ipynb
Objective	To graphically compare the different CNN architecture
Calling Function	-
Folders accessed	-

PJPTERNB#1	model_inception.h5
Objective	The model that is trained to classify the images.
Calling Function	-
Folders accessed	-

5. Annexure

5.1 Screens

Number	**Description**
SRC01	Upload Image Screen
SRC02	Result Screen

5.2 Navigation

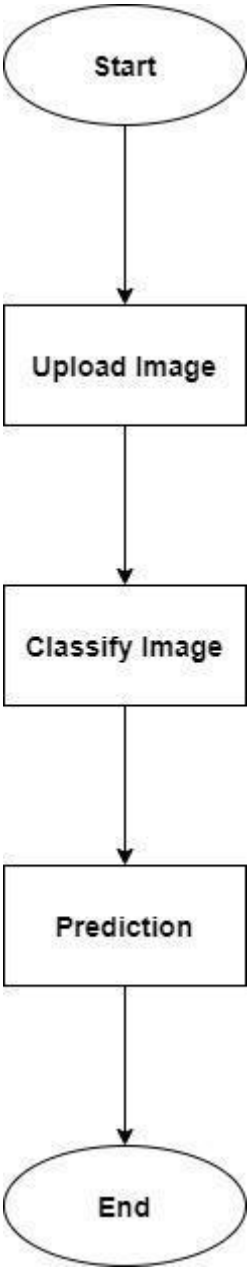

Figure 8: System Flowchart

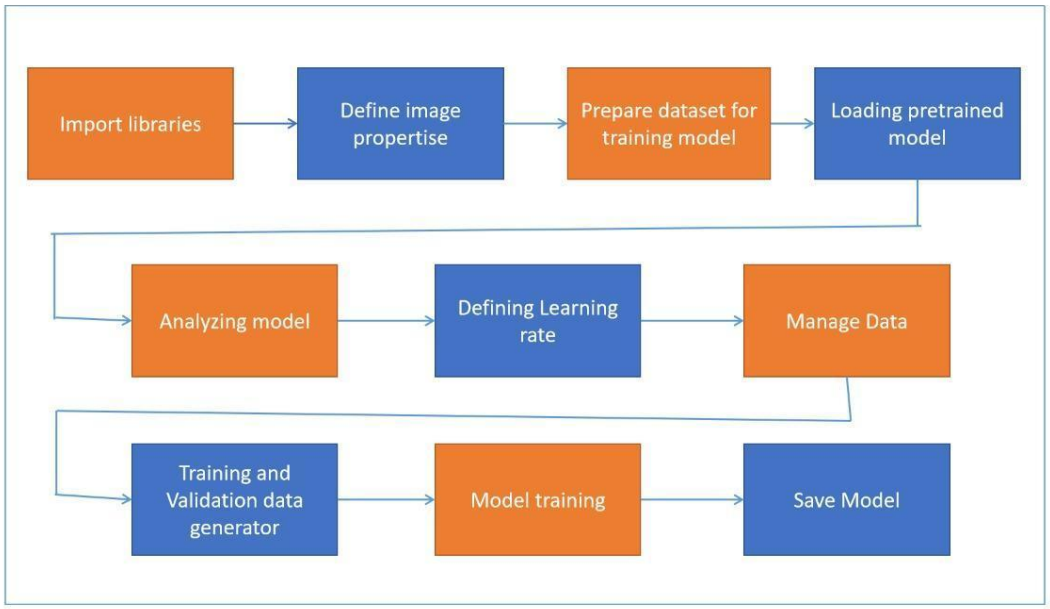

Figure 9: Model Flowchart

Image Based Plant Disease Detection

User Manual

Table Of Contents

Table Of Figures

1. INTRODUCTION

1.1 Audience

The audience of this document will be the researchers, students and teachers of any designated faculty.

1.2 Purpose of this document

This document provides help to the user in installation and efficient use of the software.

1.3 Document Organization

This document has 6 sections.

Section 1	Focuses on introduction, specifies the intended audience, purpose of this document, document organization and the contact address for assistance regarding operations of this application.
Section 2	Highlights the salient features of the application.
Section 3	Briefs about the operating conditions and handling instructions.
Section 4	Gives installation instructions.
Section 5	Provides step by step operation instructions.
Section 6	Specifies glossary.

1.4 References

The following references are considered to prepare this document.

Sr No	Title	Publisher/Author	Version	Release Date
1	SAMPARK SRS	NIC Mumbai	1.1	
2	SAMPARK SDD	NIC Mumbai	1.1	
3	SRS Template(NIC-TPL-002)	NIC Delhi	1.1	02/03/04
4	Project Life Cycle (NIC-PLC)	NIC Delhi	1.1	01/03/04
5	Quality Manual(NIC-QM)	NIC Delhi	1.0	29/02/04

1.5 Problem Reporting

In any case of query or problem please contact (**Phone/Email**) :
8779940813 / omkar.dalvi_19@sakec.ac.in
8879350579 / rummaan.ahmad_19@sakec.ac.in
8169010828 / manish.gupta_19@sakec.ac.in
9892191289/ rahul.parande_19@sakec.ac.in

2. Product Features

- The software developed is easy to operate.
- The software is designed to run on any machine having Windows or Mac Operating System with a Python code editor installed on it.
- The User can upload a picture of any extension into the User Interface without any complications.
- The model that classifies the leaves is already pre-trained and is available as model_inception.h5 and should be downloaded for proper working
- The User Interface language used is English.

3. Handling Instructions

The Software can be accessed without internet connection. The following files should be available in the folder:

1. gui.py
2. model_inception.h5
3. model1.py
4. split.py
5. graph.ipynb

4. Installation Requirements

4.1 Hardware Requirements

Windows / Mac device with appropriate Python IDE

4.2 Software Requirements

Windows / Mac with python IDE (Pycharm, VS Code, Vim, Pydev, Sublime Text, etc).

4.3 Installation Procedure

The procedure for complete installation is explained in the following steps:

1. Visit either of the below given Github repository.
   https://github.com/omkarpyc/Image-Based-Plant-Disease-Detection.git
   https://github.com/Rummaan/Image-Based-Plant-Detection-Guava
   https://github.com/R4hul04/Image-Based-Plant-Disease-Detection

2. Download the code.
3. Open the code via a Python Code Editor.
4. Install the required libraries from requirement.txt.
5. Execute the code.

5. General / Common Operating Instructions

5.1 First Screen

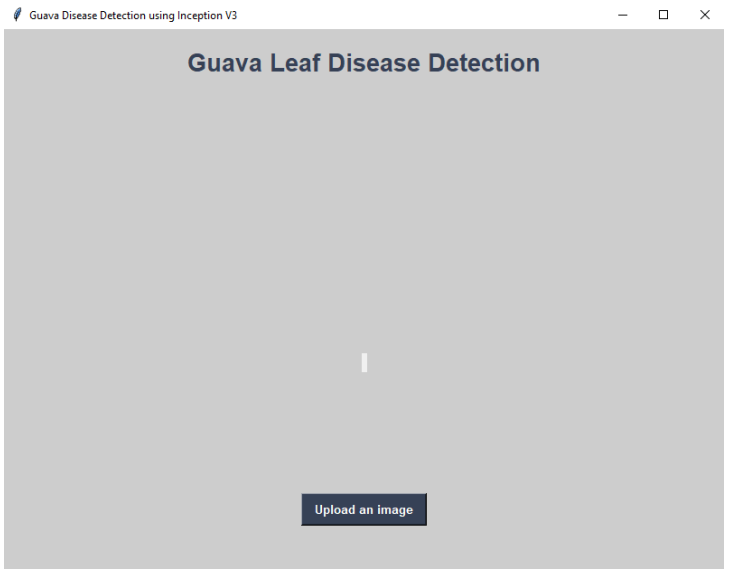

Figure 1: First Screen

5.2 Image Upload Screen

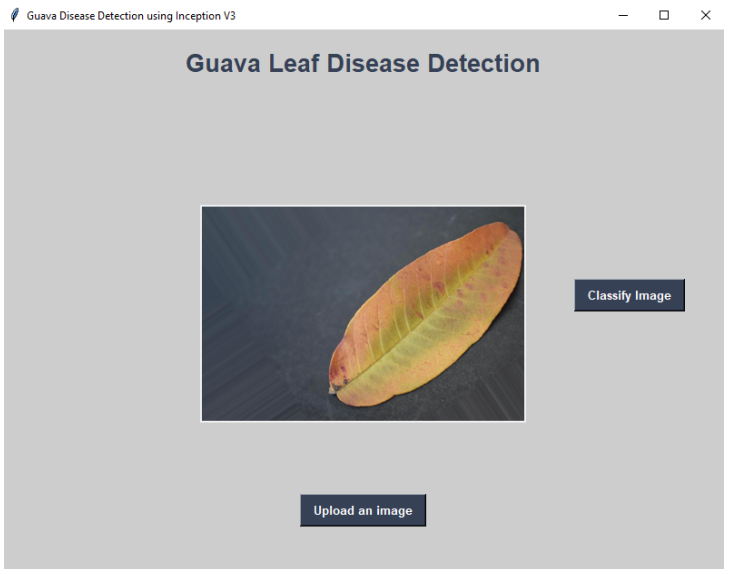

Figure 2: Image Upload Screen

5.3 After Classification Screen (Rust)

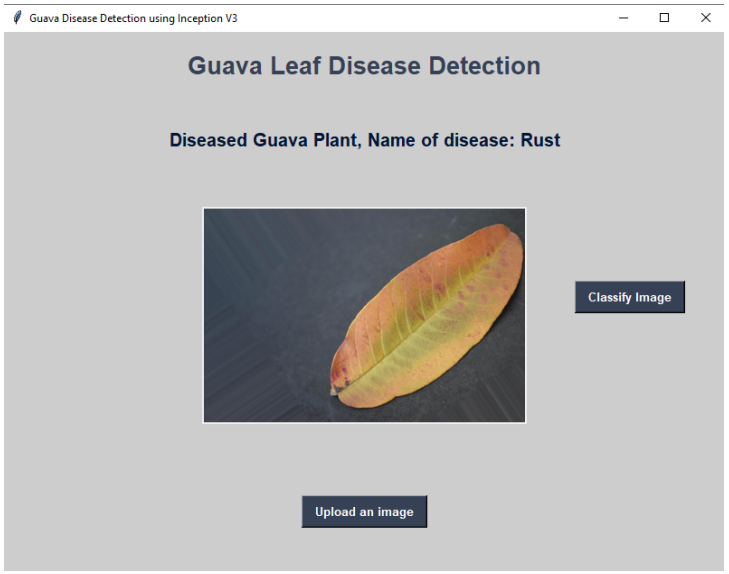

Figure 3: After Classification Screen (Rust)

5.4 After Classification Screen (Canker)

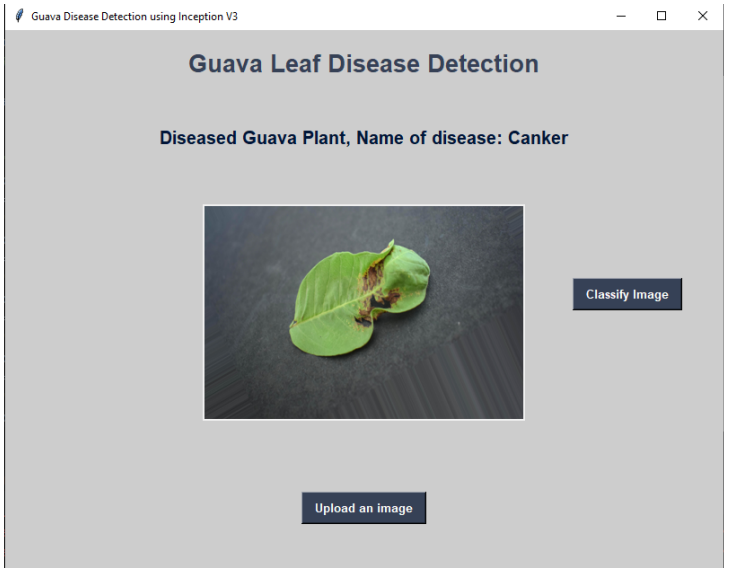

Figure 4: After Classification Screen (Canker)

5.5 After Classification Screen (Dot)

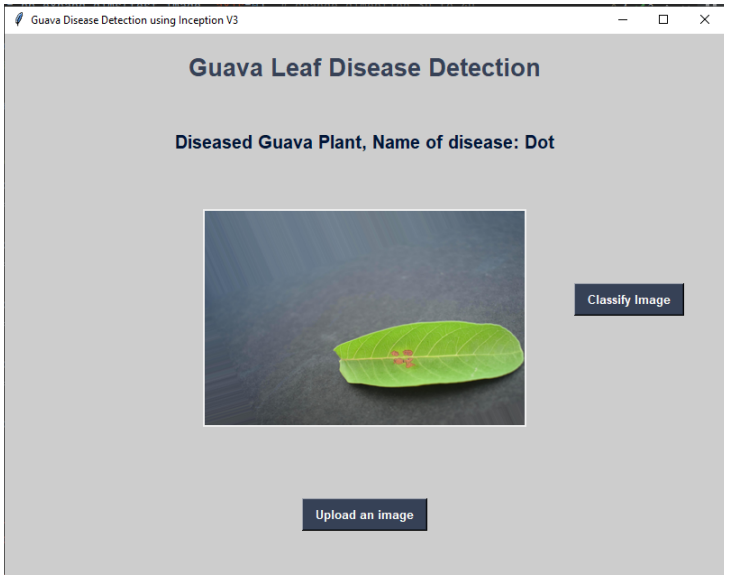

Figure 5: After Classification Screen (Dot)

5.6 After Classification Screen (Mummification)

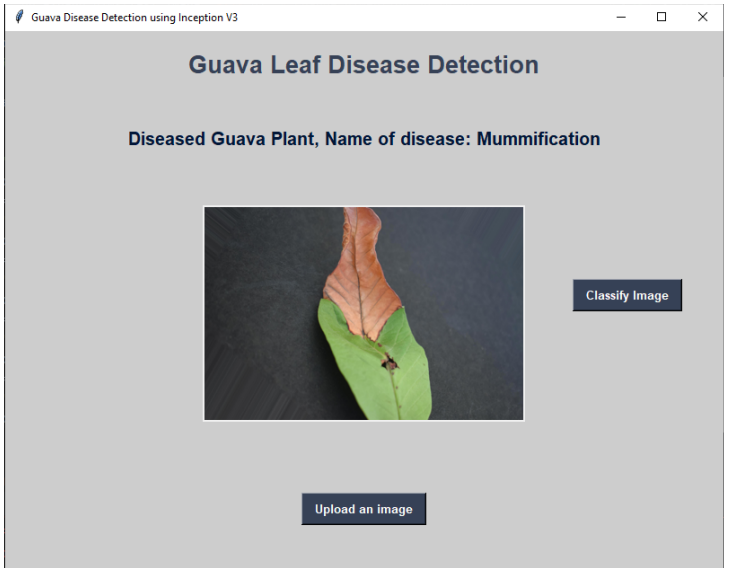

Figure 6: After Classification Screen (Mummification)

5.7 After Classification Screen (Healthy)

Figure 7: After Classification Screen (Healthy)

6. Annexure

6.1 Glossary

Sr No.	Abbreviation	Description
1	NIC	Human Resource Management System
2	MHSC	Maharashtra State Center
3	GAD	General Administrative Department
4	UM	User Manual
5	SRS	Software Requirement Specification
6	SDD	Software Design Document