INTEGRATING AI AND IMAGE ANALYSIS FOR PEST SURVEILLANCE

A PROJECT REPORT

Submitted to

Jawaharlal Nehru Technological University Kakinada, Kakinada

in partial fulfillment for the award of the degree of

Bachelor of Technology

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by
B. Aishwarya (20KN1A0517),
D. Swetha Sri Priya (20KN1A0542),
G. Abhinav Reddy (20KN1A0545)
G. Prabhu Suhas Chowdary (20KN1A0550)

Under the esteemed guidance of

Dr. D. Suneetha
Professor & HOD, CSE Department

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NRI INSTITUTE OF TECHNOLOGY
Autonomous
(Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada)
Accredited by NBA (CSE, ECE & EEE), Accredited by NAAC with 'A' Grade
ISO 9001: 2015 Certified Institution
Pothavarappadu (V), (Via) Nunna, Agiripalli (M), Eluru Dist, PIN: 521212, A.P, India.

2020-24

NRI INSTITUTE OF TECHNOLOGY

(AUTONOMOUS)

Approved by AICTE, New Delhi: Permanently Affiliated to JNTUK, Kakinada
Accredited by NAAC with "A" GRADE, Accredited by NBA (CSE, ECE&EEE)
An ISO 9001:2015 Certified Institution
Pothavarappadu (V), Agiripalli (M), Eluru District, A.P., India, Pin: 521 212
URL: www.nriit.edu.in, email: principal@nriit.edu.in, Mobile: + 91 8333882444

Certificate

This is to certify that the Project entitled **“INTEGRATING AI AND IMAGE ANALYSIS FOR PEST SURVEILLANCE”** is a bonafide work carried out by **B. Aishwarya (20KN1A0517), D. Swetha Sri Priya (20KN1A0542), G. Prabhu Suhas Chowdary (20KN1A0550), G. Abhinav Reddy (20KN1A0545)** in partial fulfillment for the award of degree of Bachelor of Technology in **Computer Science and Engineering** of **Jawaharlal Nehru Technological University Kakinada,** during the year 2020-24.

Project Guide &
Head of the Department

EXTERNAL EXAMINER

(Dr. D. SUNEETHA) &
(Dr. D. SUNEETHA)

DECLARATION

I hereby declare that the project report titled “**INTEGRATING AI AND IMAGE ANALYSIS FOR PEST SURVEILLANCE**” is a bonafide work carried out in the Department of Computer Science and Engineering, **NRI Institute of Technology, Agiripalli, Vijayawada,** during the academic year 2023-2024, in partial fulfilment for the award of the degree of **Bachelor of Technology** by JNTU Kakinada.

I further declare that this dissertation has not been submitted elsewhere for any Degree.

B. Aishwarya (20KN1A0517)

D. Swetha Sri Priya (20KN1A0542)

G. Prabhu Suhas Chowdary (20KN1A0550)

G. Abhinav Reddy (20KN1A0545)

ACKNOWLEDGEMENT

We take this opportunity to thank all who have rendered their full support to my work. The pleasure, the achievement, the glory, the satisfaction, the reward, the appreciation and the construction of my project cannot be expressed with a few words for their valuable suggestions.

We are grateful to my Project Guide **Dr. D. Suneetha, Professor** for rendering the valuable suggestions and continuous guidance to complete the project successfully.

We are grateful to my Head of the Department, for giving me helpful advice and constant support throughout the endeavour.

We extend our sincere thanks to **Dean of the Department, Dr. K. V. Sambasiva Rao** for his continuous guidance and support to complete my project successfully.

We are thankful to the **Principal, Dr. C. Naga Bhaskar** for his encouragement to complete the Project work.

We extend our sincere and honest thanks to the **Chairman, Dr. R. Venkata Rao** & **Secretary, Sri K. Sridhar** for their continuous support in completing the Project work.

Finally, We thank the Administrative Officer, Staff Members, Faculty of Department of CSE, NRI Institute of Technology and our friends, directly or indirectly helped me in the completion of this project.

20KN1A0517
20KN1A0542
20KN1A0550
20KN1A0545

Abstract

Early pest detection is a major challenge in agriculture field. The easiest way, to control the pest infection is the use of pesticides. But the excessive use of pesticides are harmful to plants, animals as well as human beings. Integrated pest management combines biological and physical methods to prevent pest infection. The techniques of machine vision and digital image Processing are extensively applied to agricultural science and it have great perspective especially in the plant protection field, which ultimately leads to crops management.

This paper deals with a new type of early detection of pest’ s system. Images of the leaves affected by pests are acquired by using a digital camera. The leaves with pest images are processed for getting a gray colored image and then using feature extraction, image classification techniques to detect pests on leaves. The images are acquired by using a digital camera. The images are then transferred to a PC and represented in python software.

The RGB image is then converted into gray scale image and the feature extraction techniques are applied on that image. The Support Vector Machine classifier is used to classify the pest types.

INDEX

CHAPTER-1
INTRODUCTION

1. Introduction

India is an agricultural country. 70 percent of the people mainly depends upon agriculture. So, increasing the productivity of crops is an important matter now. Most of the scientists are doing their researches on this field. By using their new techniques and practical implementations this is very easy. But one of the most important problems now exists is “pest infection” on plants. This paper mainly focuses on greenhouse crops. There are different crops cultivated under greenhouse. for example, vegetables like cucumber, potato, tomato etc. and flower plants like rose, jasmine etc. The most common pests which will effect on this green-house crops are whiteflies, aphids, and thrips. One way to control the pest infection is by using the pesticides. Pesticides will suppress particular species of pests. Pesticides are detrimental for the environment and produce considerable damage to eco systems.

The excessive use of pesticides will pollute air, water, and soil. Carried by the wind pesticides suspensions contaminate other areas. In this paper, we focus on early pest detection. This implies to regular observation the plants. Images are acquired using cameras. Then the acquired image has to be processed to interpret the image contents by image processing methods. The focus of this paper is on the interpretation of image for pest detection.

CHAPTER-2
LITERATURE SURVEY

2. Literature Survey

In their work, "A Learning Approach for Adaptive Image Segmentation," Vincent Martin and Monique Thonnat tackle the challenges of image segmentation in computer vision. They propose a method that automatically adjusts segmentation techniques to different contexts, addressing issues like algorithm selection and parameter tuning. By using supervised learning, their approach improves segmentation accuracy across various tasks, including figure-ground segmentation in video surveillance and static image segmentation. This project aims to enhance image understanding and boost the performance of computer vision systems.

In their study, Martin and Moisan discuss how to spot pests early in greenhouses using video analysis. They explain that quick action is crucial to stop pests from causing serious harm to plants. They introduce a system called DIViNe that uses cameras to track pests without harming the plants. This system helps farmers identify and count pests like white flies and aphids. It also learns to recognize insect behaviors. Martin and Moisan stress the importance of keeping a close eye on pests in greenhouses and making fast decisions to protect crops. Their work offers a promising solution for farmers to monitor and manage pests effectively.

In this paper, Patil and Kumar explore the application of image processing techniques in the study of plant diseases and traits, particularly within the agricultural context of India. With a significant portion of the population reliant on agriculture for livelihood, the need for technological advancements to enhance crop yield and quality is emphasized. The authors highlight the potential of image processing methods to overcome limitations associated with manual disease assessment, offering increased accuracy and efficiency in disease management. Historical examples, such as the late blight of potato in Ireland, underscore the economic impact of plant diseases and emphasize the urgency of implementing advanced technological solutions in agriculture.

In the paper "On-line counting of pests in a greenhouse using computer vision" by Ikhlef Bechar and Sabine Moisan, the authors tackle the challenge of detecting harmful insects in greenhouse crops early on. They propose a clever method that uses computer vision to count these pests in real-time. By analyzing videos, their system can identify insects even in challenging conditions like low resolution and poor lighting. They tested their approach with videos of whiteflies and found it to be very accurate, with very few mistakes. This technology could be a game-changer for farmers, helping them protect their crops more effectively.

The paper "A cognitive vision approach to early pest detection in greenhouse crops" by Paul Boissard, Vincent Martin, and Sabine Moisan introduces a new method for spotting pests in greenhouse plants early on. Traditional methods for checking for pests in plants are slow and not very accurate. To solve this problem, the authors suggest using computers and artificial intelligence to automatically scan plant leaves and identify any pests, like whiteflies. They tested their system on rose leaves and found that it worked well, even when there were only a few pests present. This approach could help farmers catch pests sooner and protect their crops better. The authors plan to improve the system further and use it to detect other types of pests and plant diseases in the future.

B. Cunha's project "Application of Image Processing in Characterization of Plants" explores the use of image processing techniques to analyze plants. This research follows a growing trend in using image analysis to understand plant growth and health. By leveraging advances in image processing, the project aims to develop innovative methods for studying plants. The ultimate goal is to enhance agricultural practices and environmental science through more efficient and accurate plant analysis.

Detection of Pests Using Video Analysis:

This work combines image processing techniques as well as knowledge-based technique. It will detect only whiteflies. The result of this system is more reliable and accurate than that of the manual methods. This is actually a multidisciplinary cognitive vision system that combines different types of techniques like computer vision, artificial intelligence, image processing etc. In this work, they chose rose plant as the testing crop and white fly as the pest for testing. The early stage of detection was quite difficult. So, they chose adult flies. But some problems were there in detection of adult also. The adult may fly away during the image capturing time. So, they chose to

scan the leaves of rose when flies were not active. The future scope of the work is to detect whiteflies in its early stage.

Method which uses Sticky Traps

The goal of Detection of insects by a video camera network is to detect the pest infection on leaves by using a video analysis. The traditional methods will take more time to detect and count the pests. Because of this reason they have developed an automatic system based on video analysis. They used 5 wireless cameras in greenhouse. They chose rose as a crop for testing. sticky traps are used in this work. Sticky traps are nothing but a sticky material which is having some colors to attract the pests. For the detection of insects, they used video segmentation algorithms with learning and adaptation techniques. The adaptive system can be used in any weather conditions. The future scope of this system is to detect new types of pests in early stage.

2.1 Existing System

The methods which are used for the early detection of pests in green house crops along with their advantages and disadvantages. The methods are explained below with their features and drawbacks.

Detection of Pests Using Video Analysis

This work combines image processing techniques as well as knowledge-based technique. It will detect only whiteflies. The result of this system is more reliable and accurate than that of the manual methods. This is actually a multidisciplinary cognitive vision system that combines different types of techniques like computer vision, artificial intelligence, image processing etc.

Method which use Sticky Traps

The goal of Detection of insects by a video camera network is to detect the pest infection on leaves by using a video analysis. The traditional methods will take more time to detect and count the pests. Because of this reason they have developed an automatic system based on video analysis. They used 5wirelesscamerasin greenhouse.

Disadvantages:

- ♦ Time Consuming
- ♦ Need More Man Power
- ♦ Difficult to handle
- ♦ Difficult to identify pest disease

2.2 Proposed System

For this study, whiteflies and aphids are chosen because this pest requires early detection and treatment to prevent durable infection. Samples are collected by using the pan-tilt camera with zoom in green house. The acquired Images are given to the local machine and the image processing techniques will takes place.

Advantages:

- Consume Less Time
- Minimal Manpower
- Paper less management
- Hassle free
- Easy to Maintain

2.3 System Requirements

HARDWARE REQUIREMENTS:

- **System** : MINIMUM i3.
- **Hard Disk** : 40 GB.
- **RAM** : 4 GB.

SOFTWARE REQUIREMENTS:

- **Operating System:** Windows 8
- **Coding Language**: Python 3.7

2.4 System Study

Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- **ECONOMICAL FEASIBILITY**
- **TECHNICAL FEASIBILITY**
- **SOCIAL FEASIBILITY**

Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER-3
METHODOLOGY

3. Methodology

Image capturing

The first step of every image processing application is image acquisition or image capturing. The images of leaves are captured by using the camera and it will store it in some formats like .PNG, .JPG, .JPEG etc.

Image pre-processing

Image preprocessing is used to create an enhanced and please full version of the captured image. The image preprocessing steps used in the system are:

1) Conversion of RGB image to gray image
2) Resizing of the image
3) Filtering of the image.

a) Conversion of RGB to Gray Image

In RGB color model, each color appears in its primary spectral components of red, green, and blue. The color of a pixel is made up of three components; red, green, and blue (RGB). The disadvantages of RGB models are, it requires large space to store and it will take more time to process. So, there is a need for converting the RGB model to Gray model.

b) Resizing of the Image

Resizing is an important step in image preprocessing. The acquired image is resized according to the requirement of the system. Resizing of the image: Resizing is nothing but, changing the dimensions of an image. The captured image is resized using some resizing methods according to the requirement of the system. There are different methods for the resizing of images. Blinear, Bicubic and Nearest neighborhood interpolation are the common resizing methods. Here in our system, we are using bicubic method.

c) Filtering of the image

Filtering is nothing but, eliminating the unwanted portion of the image. Different types of filters are available. Low pass filters are smoothening filters, it will pass only low frequency signals and eliminate all the high frequency signals. High pass filters are sharpening filters, and it will eliminate all the low frequency signals and pass only high frequency signals. Band pass filters will pass the signals which is having a specific range of frequencies.

In our system we are using smoothening filter. The purpose of smoothing is to reduce noise and improve the visual quality of the image. Spatial filters are applied to both static and dynamic images, whereas temporal images are applied only to dynamic images. The simplest smoothening filter is average filter. It consists of a 3X3 matrix of 1 and it is divided by 9.

Feature Extraction

Feature extraction is the most important part of this project. Some properties of the images are considered here. The different types of properties include region properties, gray covariance matrix properties etc. The properties standard deviation, entropy, contrast etc are extracted from the image and are used to train the dataset for the SVM classification. Support Vector Machines (SVM's) are a relatively new learning method used for binary classification. The basic idea is to find a hyper plane which separates the d-dimensional data perfectly into its two classes. The different types of properties of an image is listed in the table below.

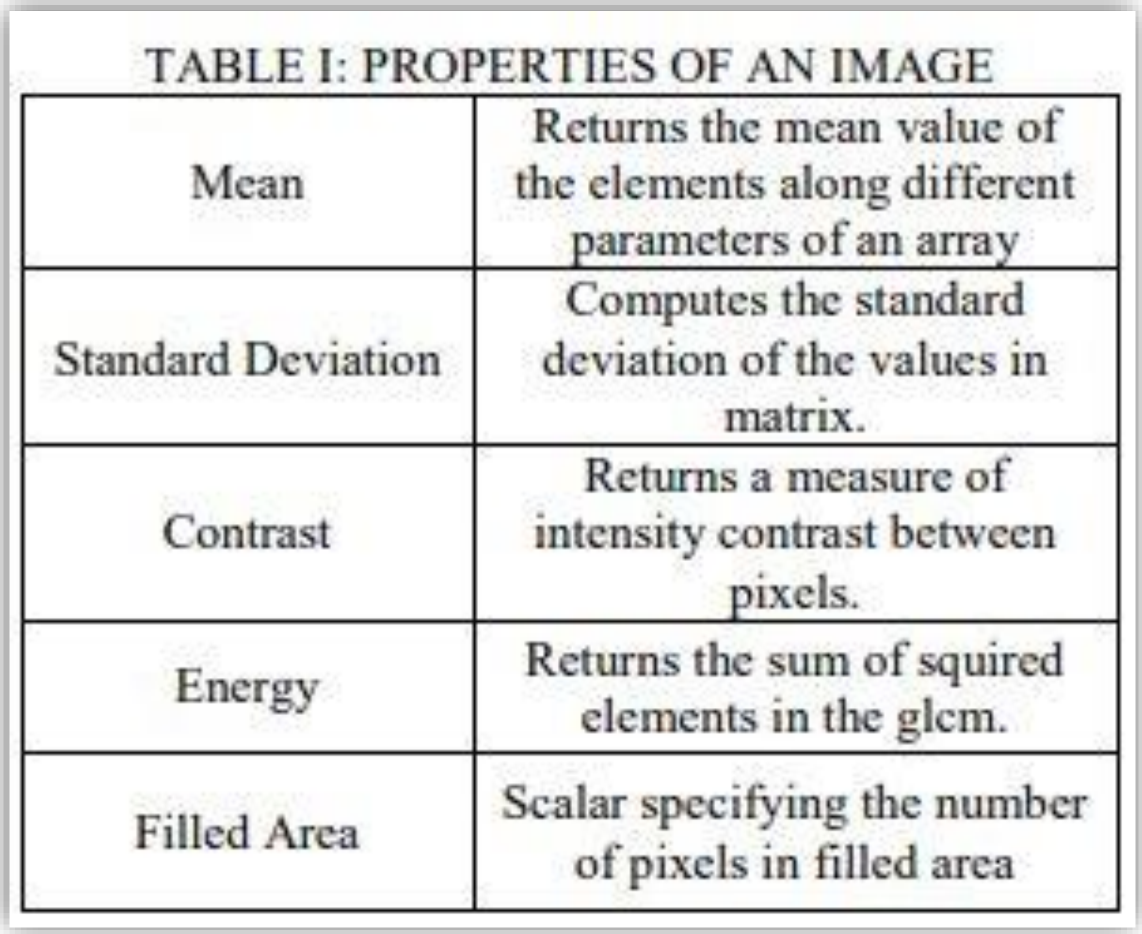


TABLE I: PROPERTIES OF AN IMAGE

Mean	Returns the mean value of the elements along different parameters of an array
Standard Deviation	Computes the standard deviation of the values in matrix.
Contrast	Returns a measure of intensity contrast between pixels.
Energy	Returns the sum of squired elements in the glcm.
Filled Area	Scalar specifying the number of pixels in filled area

Fig :3.1 Properties of an Image

Detection and Classification

In this module the affected and unaffected images are compared by using the dataset provide in the SVM. If it is an affected image again it is compared by using the second dataset provided in the SVM. From this comparison the type of pest can be detected.

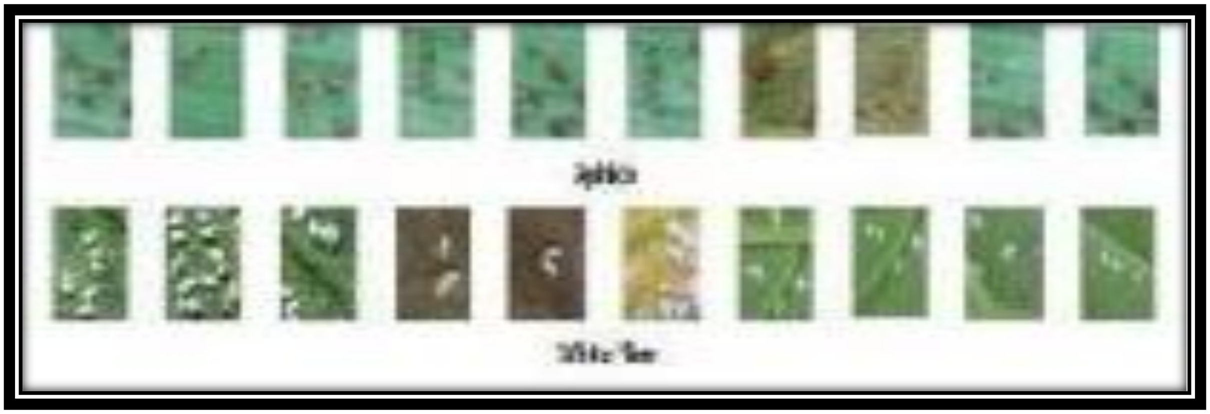


Fig:3.2 Detection & Classification

Flowchart

Flowchart for the proposed system is given in figure. The images are acquired by using camera and it is filtered by using bicubic filters to avoid unwanted noise portions. This is actually the image preprocessing step. The next step is svm classification to detect the pest infection. If the image is affected, then again it is applied to the svm to detect the type of pest.

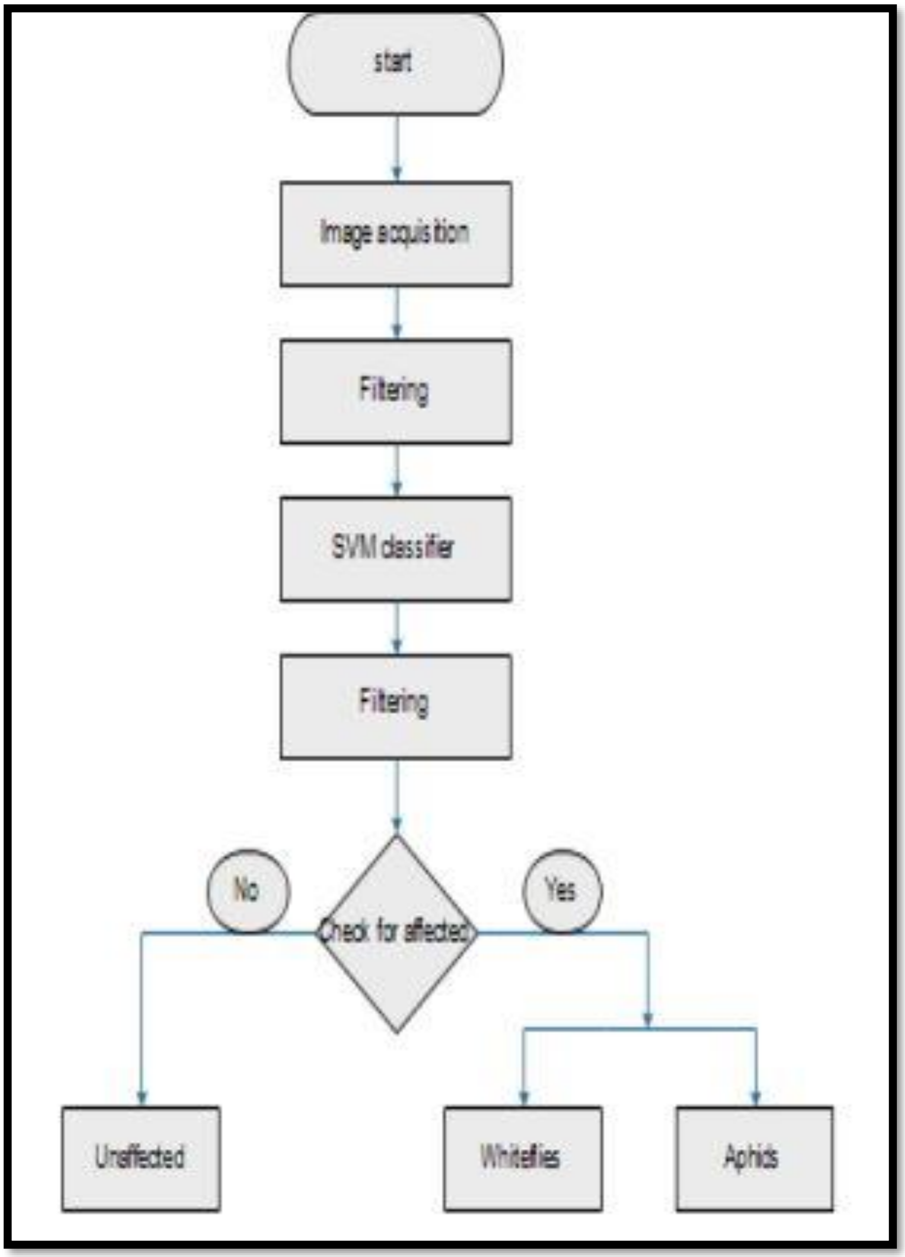


Fig:3.3 Flowchart

CHAPTER-4
SYSTEM DESIGN

4. System Design

4.1 UML Diagrams:

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

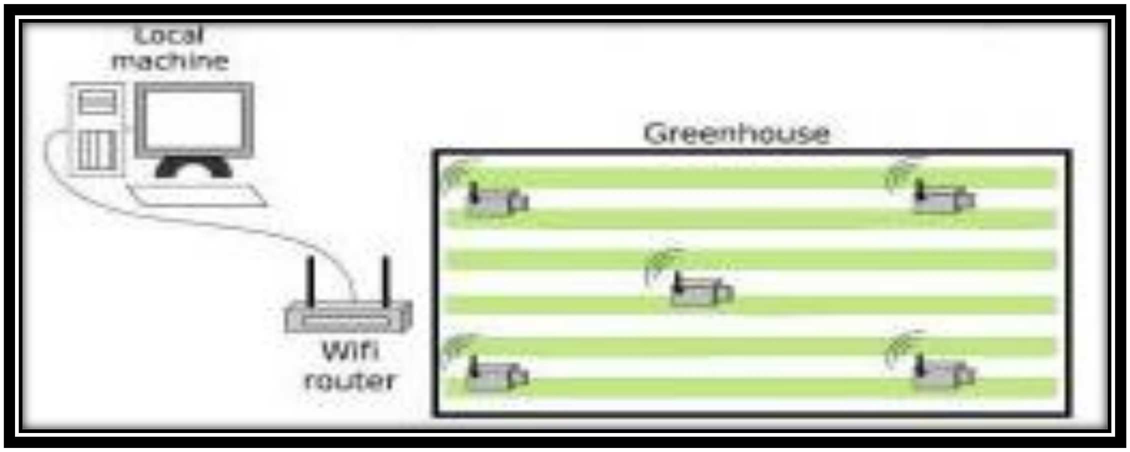
Goals:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns, and components.
7. Integrate best practices.

4.1.1 Proposed Model

For this study, whiteflies and aphids are chosen because this pest requires early detection and treatment to prevent durable infection. Samples are collected by using the pan tilt camera with zoom in greenhouse as shown in Fig.1.The acquired Images are given to the local machine and the image processing techniques will takes place.



System Architecture

Fig:4.1 System Architecture

4.1.2 Use Case Diagram:

A use case diagram in 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.

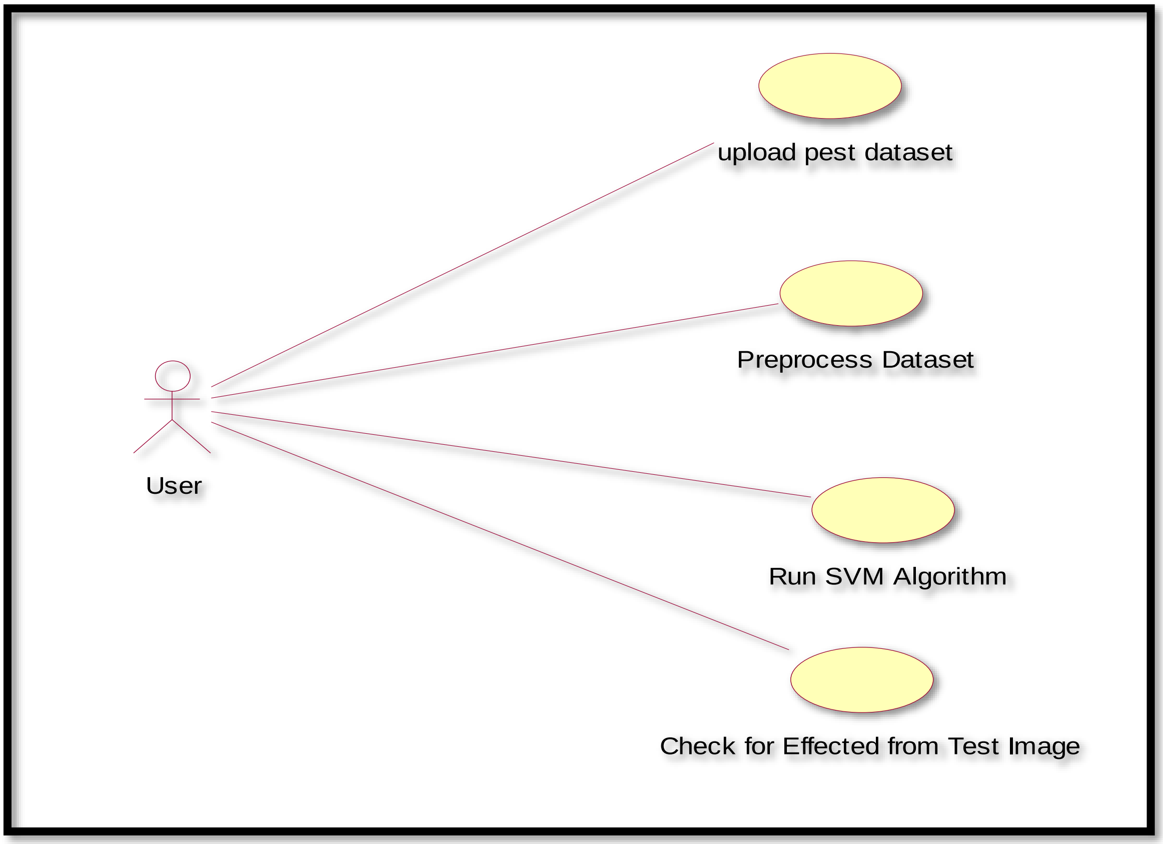


Fig:4.2 Use Case Diagram

4.1.3 Class Diagram:

In software engineering, a class diagram in 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.

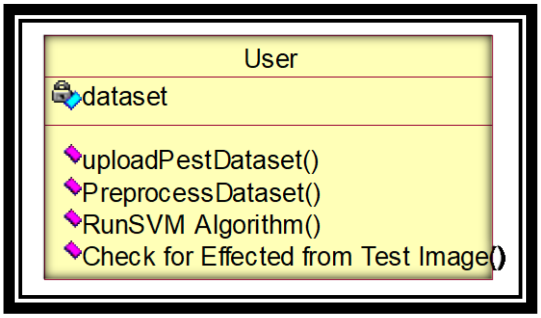


Fig:4.3 Class Diagram

4.1.4 Sequence Diagram:

A sequence diagram in 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. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

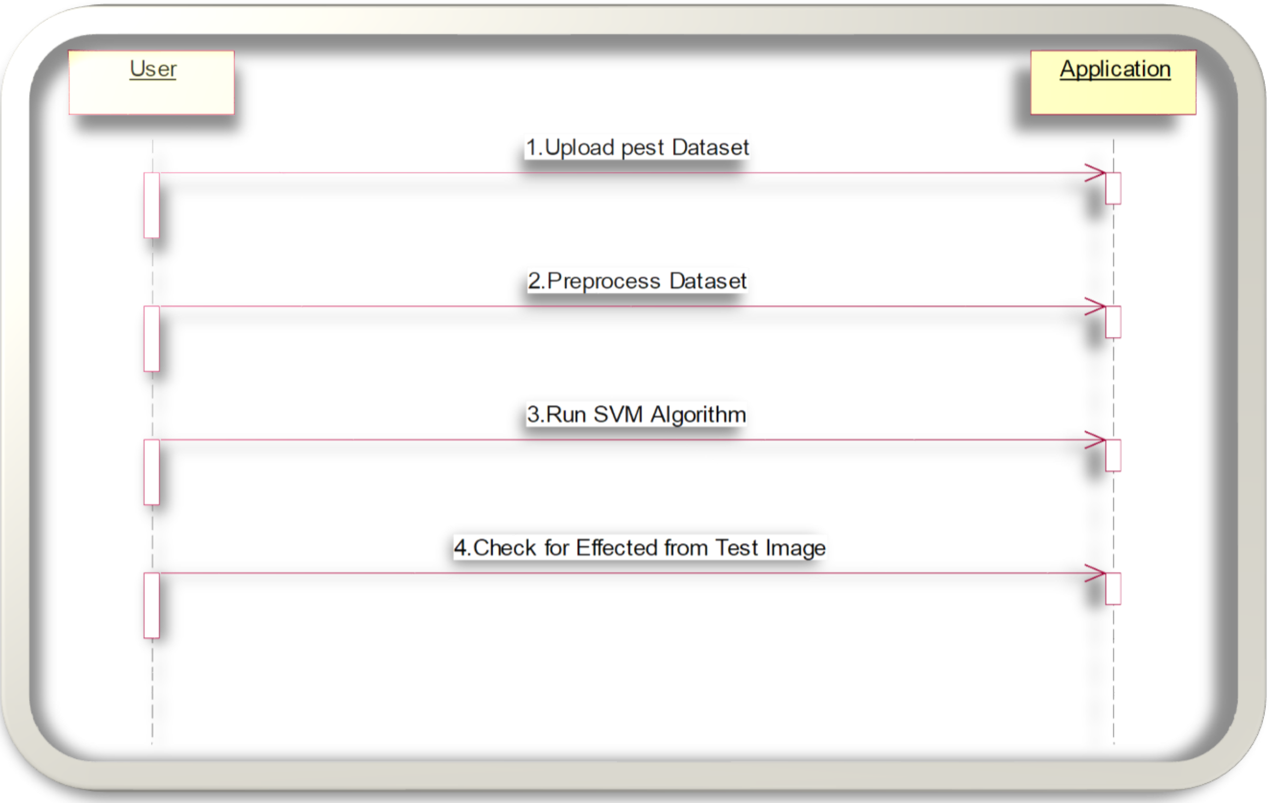


Fig:4.4 Sequence Diagram

4.1.5 Collaboration Diagram:

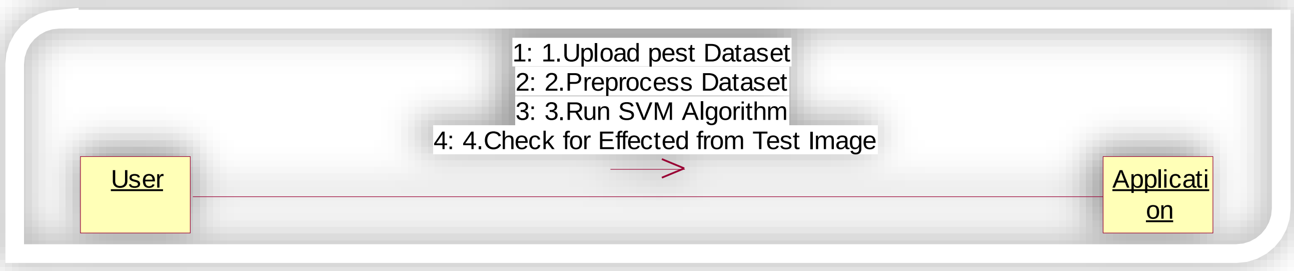


Fig:4.5 Collaboration Diagram

4.2 Implementation:

Modules:

1) **Upload Pest Dataset:**

using this module, we will upload dataset to application.

2) **Preprocess Dataset:**

using this module, we will acquire images from dataset and then filter images to grey color and then normalize images and then split dataset into train and test part where application use 80% images for training and 20% for testing

3) **Run SVM Algorithm:**

process images will be input to SVM algorithm for training and then calculate its prediction accuracy.

4) **Check for Effected from Test Image:**

using this module, we will upload test image and then SVM will predict type of pest as Aphid, White fly or Unaffected.

CHAPTER-5
SOFTWARE ENVIRONMENT

5. Software Environment

What is Python: -

Below are some facts about Python.

Python is currently the most widely used multi-purpose, high-level programming language.

Python allows programming in Object-Oriented and Procedural paradigms. Python program generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard libraries which can be used for the following –

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like Opencv, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia

Advantages of Python: -

Let us see how Python dominates over other languages.

1. **Extensive Libraries**

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. *So, we do not have to write the complete code for that manually.*

2. **Extensible**

As we have seen earlier, Python can be **extended to other languages**. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

3. **Embeddable**

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add **scripting capabilities** to our code in the other language.

4. **Improved Productivity**

The language's simplicity and extensive libraries render programmers **more productive** than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

5. **IOT Opportunities**

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

6. **Simple and Easy**

When working with Java, you may have to create a class to print **'Hello World'**. But in Python, just a print statement will do. It is also quite **easy to learn, understand,** and **code.**

This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

7. **Readable**

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and **indentation is mandatory.** These further aids the readability of the code.

8. **Object-Oriented**

This language supports both the **procedural and object-oriented** programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the **encapsulation of data** and functions into one.

9. **Free and Open-Source**

Like we said earlier, Python is **freely available.** But not only can you **download Python** for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

10. **Portable**

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn’t the same with Python. Here, you need to **code only once**, and you can run it anywhere. This is called **Write Once Run Anywhere (WORA)**. However, you need to be careful enough not to include any system-dependent features.

11. **Interpreted**

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, **debugging is easier** than in compiled languages.

Any doubts till now in the advantages of Python? Mention in the comment section.

Advantages of Python Over Other Languages

1. **Less Coding**

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you do not have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

2. **Affordable**

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

The 2019 GitHub annual survey showed us that Python has overtaken Java in the most popular programming language category.

3. **Python is for Everyone**

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and **machine learning**, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

Disadvantages of Python

So far, we have seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let’s now see the downsides of choosing Python over another language.

1. **Speed Limitations**

We have seen that Python code is executed line by line. But since Python is interpreted, it often results in **slow execution**. This, however, isn’t a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

2. **Weak in Mobile Computing and Browsers**

While it serves as an excellent server-side language, Python is much rarely seen on the **client-side**. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called **Carbon Nelle**.

The reason it is not so famous despite the existence of Bryton is that it isn't that secure.

3. **Design Restrictions**

As you know, Python is **dynamically-typed**. This means that you don't need to declare the type of variable while writing the code. It uses **duck-typing**. But wait, what's that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can **raise run-time errors**.

4. **Underdeveloped Database Access Layers**

Compared to more widely used technologies like **JDBC (Java Database Connectivity)** and **ODBC (Open Database Connectivity)**, Python's database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

5. **Simple**

No, we're not kidding. Python's simplicity can indeed be a problem. Take my example. I don't do Java, I'm more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

History of Python: -

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica).

The greatest achievement of ABC was to influence the design of Python. Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners[1], Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde Informatica (CWI). I do not know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it. "Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So, I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

What is Machine Learning: -

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it is more helpful to think of machine learning as a means of building models of data.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models *tunable parameters* that can be adapted to observed data; in this way the program can be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain. Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

Categories Of Machine Leaning: -

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into *classification* tasks and *regression* tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning involves modeling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself."

These models include tasks such as *clustering* and *dimensionality reduction.* Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate, and solve complex problems. On the other side, AI is still in its initial stage and have not surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, “to make decisions, based on data, with efficiency and scale”.

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently. The fact is that we can’t do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

Challenges in Machines Learning: -

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go.

The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are –

Quality of data − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

Time-Consuming task − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

Lack of specialist persons − As ML technology is still in its infancy stage, availability of expert resources is a tough job.

No clear objective for formulating business problems − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

Issue of overfitting & underfitting − If the model is overfitting or underfitting, it cannot be represented well for the problem.

Curse of dimensionality − Another challenge ML model faces is too many features of data points. This can be a real hindrance.

Difficulty in deployment − Complexity of the ML model makes it quite difficult to be deployed in real life.

Applications of Machines Learning: -

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML.

It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML −

- Emotion analysis
- Sentiment analysis
- Error detection and prevention
- Weather forecasting and prediction
- Stock market analysis and forecasting
- Speech synthesis
- Speech recognition
- Customer segmentation
- Object recognition
- Fraud detection
- Fraud prevention
- Recommendation of products to customer in online shopping

How to Start Learning Machine Learning?

Arthur Samuel coined the term **“Machine Learning”** in 1959 and defined it as a **“Field of study that gives computers the capability to learn without being explicitly programmed”.** And that was the beginning of Machine Learning! In modern times, Machine Learning is one of the most popular (if not the most!) career choices. According to Indeed, Machine Learning Engineer Is the Best Job of 2019 with a *344%* growth and an average base salary of **$146,085** per year.

But there is still a lot of doubt about what exactly is Machine Learning and how to start learning it? So, this article deals with the Basics of Machine Learning and also the path you can follow to eventually become a full-fledged Machine Learning Engineer. Now let’s get started!!!

How to start learning ML?

This is a rough roadmap you can follow on your way to becoming an insanely talented Machine Learning Engineer. Of course, you can always modify the steps according to your needs to reach your desired end-goal!

Step 1 – Understand the Prerequisites

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus, Statistics, and Python. And if you don’t know these, never fear! You don’t need a Ph.D. degree in these topics to get started but you do need a basic understanding.

(a) **Learn Linear Algebra and Multivariate Calculus**

Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning, then you will not be that heavily focused on math as there are many common libraries available. But if you want to focus on R&D in

Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms from scratch.

(b) **Learn Statistics**

Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection, analysis, and presentation of data. Some of the key concepts in statistics that are important are Statistical Significance, Probability Distributions, Hypothesis Testing, Regression, etc. Also, Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, and Posteriors, Maximum Likelihood, etc.

(c) **Learn Python**

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is Python! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning such as Keras, TensorFlow, Scikit-learn, etc.

So, if you want to learn ML, it's best if you learn Python! You can do that using various online resources and courses such as **Fork Python** available Free on GeeksforGeeks.

Step 2 – Learn Various ML Concepts

Now that you are done with the prerequisites, you can move on to actually learning ML (Which is the fun part!!!) It's best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in ML are:

(a) **Terminologies of Machine Learning**

- **Model –** A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.
- **Feature –** A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, etc.
- **Target (Label) –** A target variable or label is the value to be predicted by our model. For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.
- **Training –** The idea is to give a set of inputs(features) and it's expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.
- **Prediction –** Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

(b) **Types of Machine Learning**

- **Supervised Learning –** This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.
- **Unsupervised Learning –** This involves using unlabeled data and then finding the underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.
- **Semi-supervised Learning –** This involves using unlabeled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.
- **Reinforcement Learning –** This involves learning optimal actions through trial and error. So, the next action is decided by learning behaviors that are based on the current state and that will maximize the reward in the future.

Advantages of Machine learning: -

1. **Easily identifies trends and patterns -**

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

2. **No human intervention needed (automation)**

With ML, you don't need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and improve the algorithms on their own. A common example of this is anti-virus software's; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

3. **Continuous Improvement**

As **ML algorithms** gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

4. **Handling multi-dimensional and multi-variety data**

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

5. **Wide Applications**

You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

Disadvantages of Machine Learning: -

1. **Data Acquisition**

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

2. **Time and Resources**

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

3. **Interpretation of Results**

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

4. **High error-susceptibility**

Machine Learning is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

Python Development Steps: -

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt. sources in February 1991. This release included already exception handling, functions, and the core data types of list, dict, str and others. It was an object oriented.
Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked. Six and a half years later in October 2000, Python 2.0 was introduced. This

release included list comprehensions, a full garbage collector and it was supporting Unicode. Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it. "Some changes in Python 7.3:

- Print is now a function
- Views and iterators instead of lists
- The rules for ordering comparisons have been simplified. E.g. a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.
- There is only one integer type left, i.e. int. long is int as well.
- The division of two integers returns a float instead of an integer. "//" can be used to have the "old" behavior.
- Text Vs. Data Instead of Unicode Vs. 8-bit

Purpose: -

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

Modules Used in Project: -

TensorFlow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

NumPy

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object

- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using NumPy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and I Python shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits.

Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

Scikit – learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something

about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

Install Python Step-by-Step in Windows and Mac:

Python a versatile programming language doesn't come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

How to Install Python on Windows and Mac:

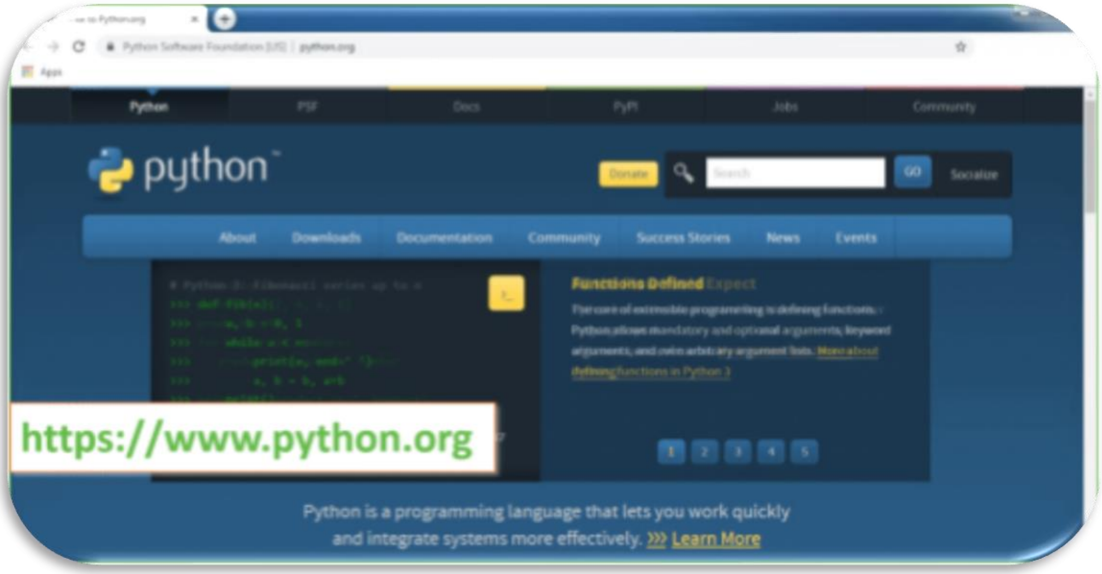
There have been several updates in the Python version over the years. The question is how to I install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

Note: The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your **System Requirements**. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a **Windows 64-bit operating system**. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. Download the Python Cheat sheet here._The steps on how to install Python on Windows 10, 8 and 7 are **divided into 4 parts** to help understand better.

Download the Correct version into the system

Step 1: Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: https://www.python.org



https://www.python.org

Functions Defined

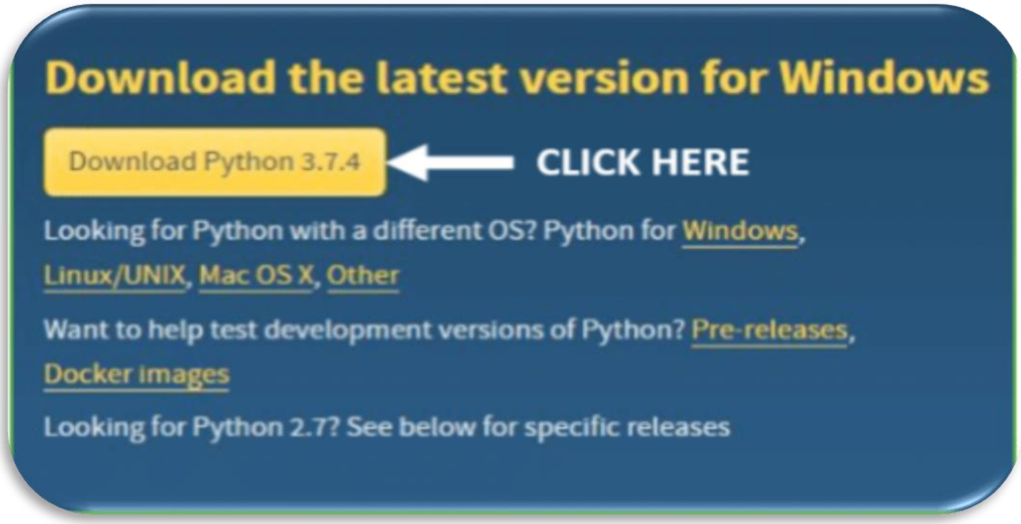
The core of extensible programming is defining functions. Python allows mandatory and optional arguments, keyword arguments, and even arbitrary argument lists. More about defining functions in Python 3

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> Learn More

Fig: 5.1

Now, check for the latest and the correct version for your operating system.

Step 2: Click on the Download Tab.



Download the latest version for Windows

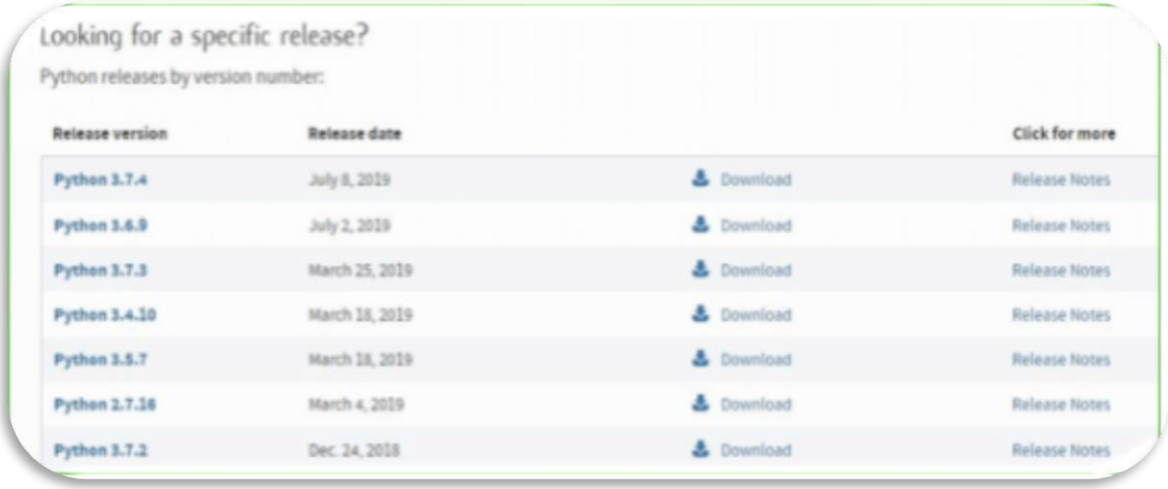
Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other

Want to help test development versions of Python? Pre-releases, Docker images

Looking for Python 2.7? See below for specific releases

Fig:5.2

Step 3: You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4.



Looking for a specific release?

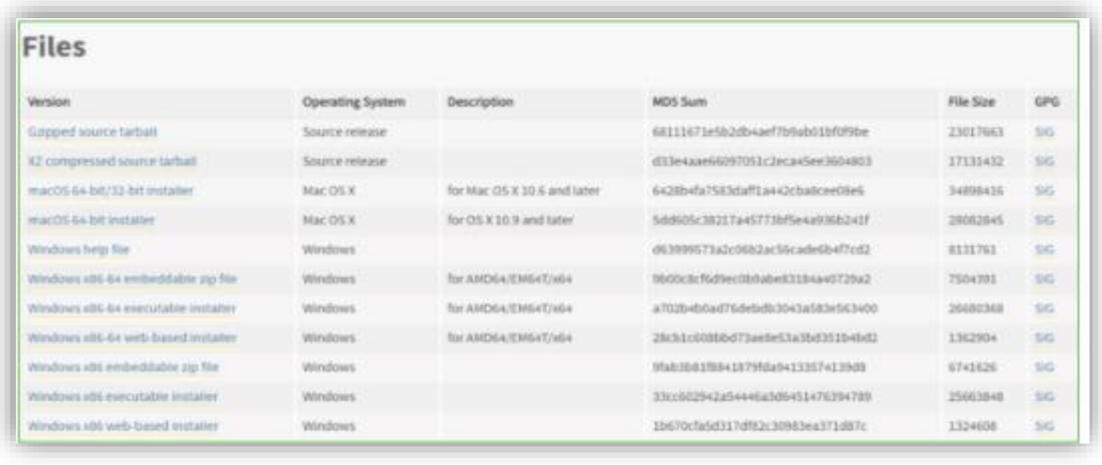
Python releases by version number:

Release version	Release date		Click for more
Python 3.7.4	July 8, 2019	Download	Release Notes
Python 3.6.9	July 2, 2019	Download	Release Notes
Python 3.7.3	March 25, 2019	Download	Release Notes
Python 3.4.10	March 18, 2019	Download	Release Notes
Python 3.5.7	March 18, 2019	Download	Release Notes
Python 2.7.16	March 4, 2019	Download	Release Notes
Python 3.7.2	Dec. 24, 2018	Download	Release Notes

Fig:5.3

Step 4: Scroll down the page until you find the Files option.

Step 5: Here you see a different version of python along with the operating system.



Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		68111671e5b2db4aef7b9ab01bf0f9be	23017663	SIG
XZ compressed source tarball	Source release		d33e4aae66097051c2eca45ee3604803	17131432	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6428b4fa7583daff1a442cbalcee08e6	34898416	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	5dd605c38217a45773bf5e4a936b241f	28082845	SIG
Windows help file	Windows		d63999573a2c06b2ac56cade6b4f7cd2	8131761	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	9b00c8cf6d9ec0b9abe83184a40729a2	7504391	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	a702b4b0ad76debdb3043a583e563400	26680368	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	28cb1c608bbd73ae8e53a3bd351b4bd2	1362904	SIG
Windows x86 embeddable zip file	Windows		9fab3b81f8841879fda94133574139d8	6741626	SIG
Windows x86 executable installer	Windows		33cc602942a54446a3d6451476394789	25663848	SIG
Windows x86 web-based installer	Windows		1b670cfa5d317df82c30983ea371d87c	1324608	SIG

Fig:5.4

- To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.

•To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

Note: To know the changes or updates that are made in the version you can click on the Release Note Option.

Installation of Python

Step 1: Go to Download and Open the downloaded python version to carry out the installation process.

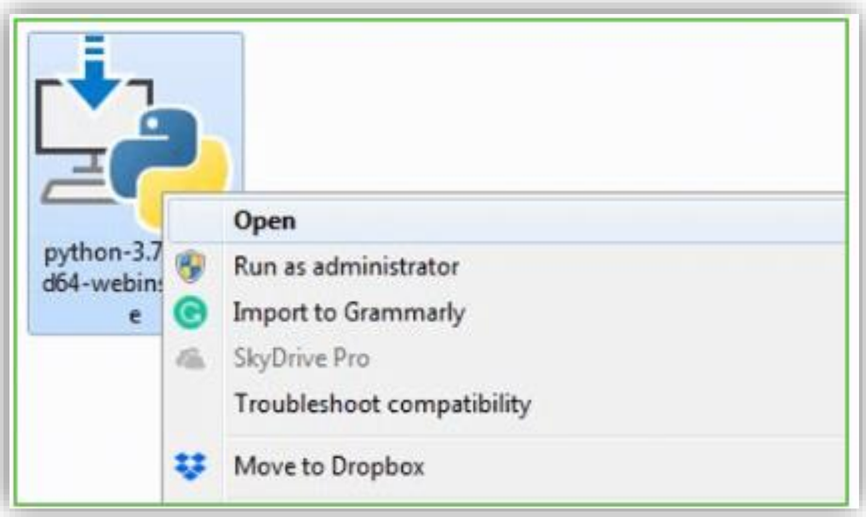
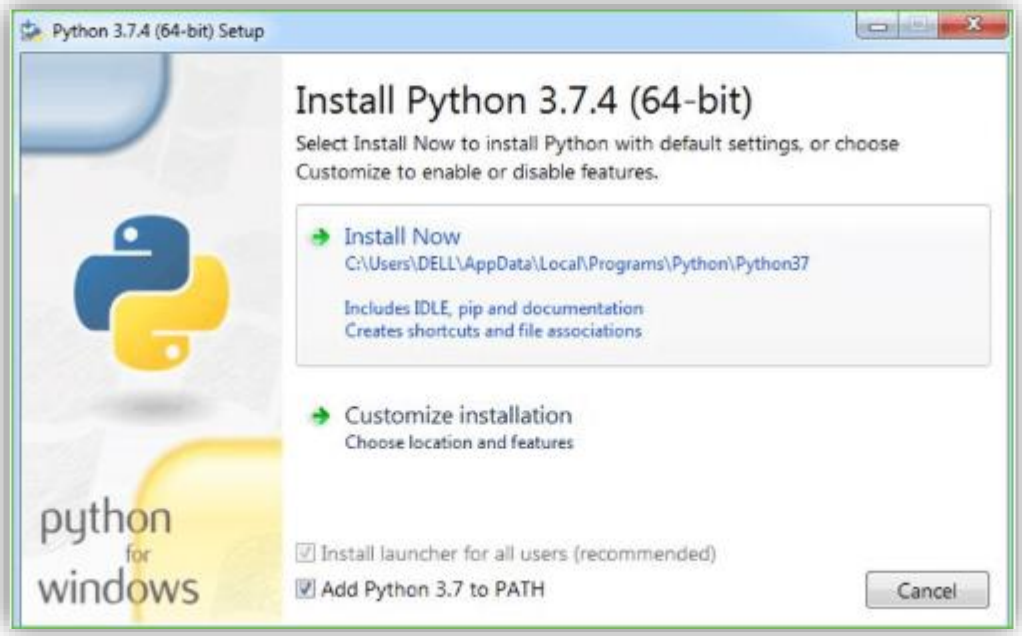


Fig:5.5

Step 2: Before you click on Install Now, make sure to put a tick on Add Python 3.7 to PATH.



Python 3.7.4 (64-bit) Setup

Install Python 3.7.4 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

➔ Install Now
C:\Users\DELL\AppData\Local\Programs\Python\Python37

Includes IDLE, pip and documentation
Creates shortcuts and file associations

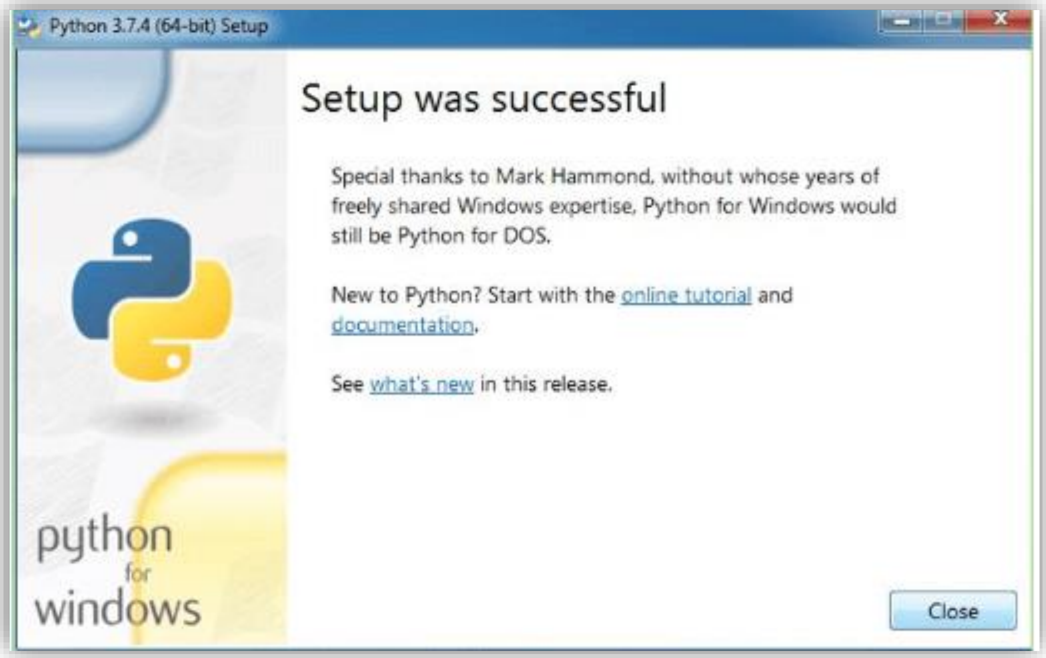
➔ Customize installation
Choose location and features

☑ Install launcher for all users (recommended)

☑ Add Python 3.7 to PATH

Fig:5.6

Step 3: Click on Install NOW After the installation is successful. Click on Close.



Python 3.7.4 (64-bit) Setup

Setup was successful

Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.

New to Python? Start with the online tutorial and documentation.

See what's new in this release.

Fig:5.7

With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

Note: The installation process might take a couple of minutes.

Verify the Python Installation

Step 1: Click on Start

Step 2: In the Windows Run Command, type “cmd”.

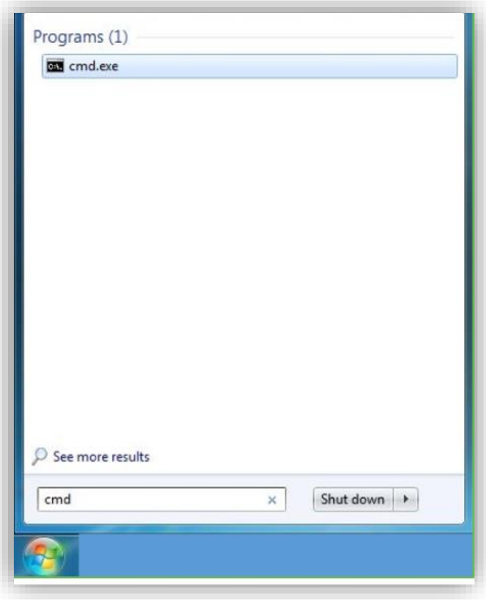
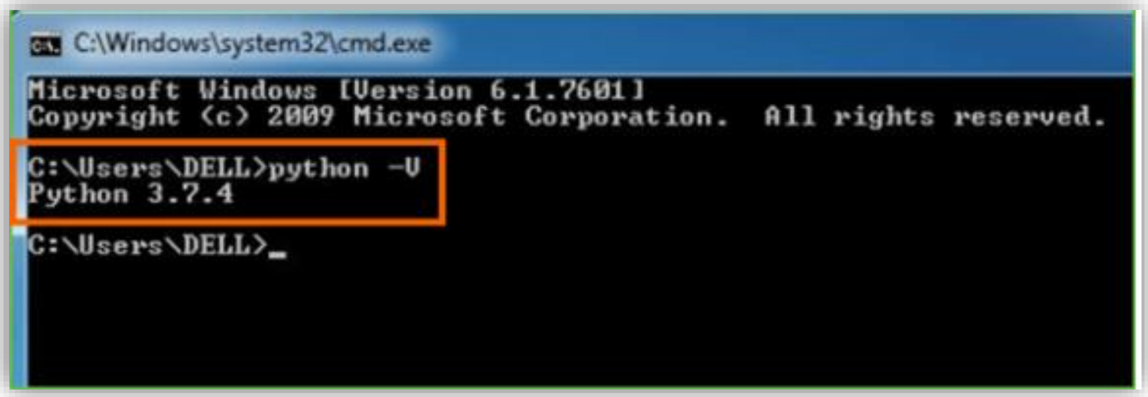


Fig:5.8

Step 3: Open the Command prompt option.

Step 4: Let us test whether the python is correctly installed. Type **python –V** and press Enter.



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\DELL>python -V
Python 3.7.4

C:\Users\DELL>_
```

Fig:5.9

Step 5: You will get the answer as 3.7.4

Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

Check how the Python IDLE works

Step 1: Click on Start

Step 2: In the Windows Run command, type "python idle".

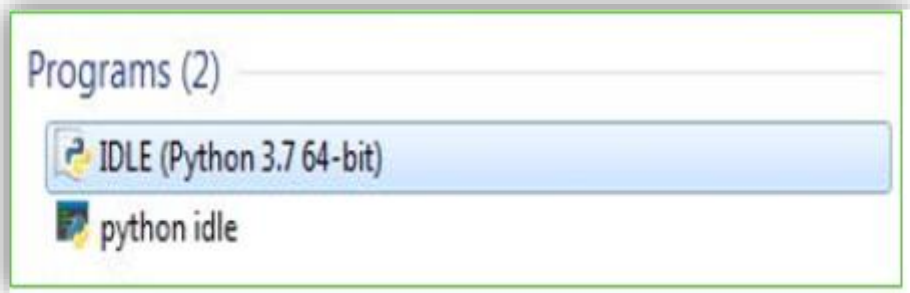
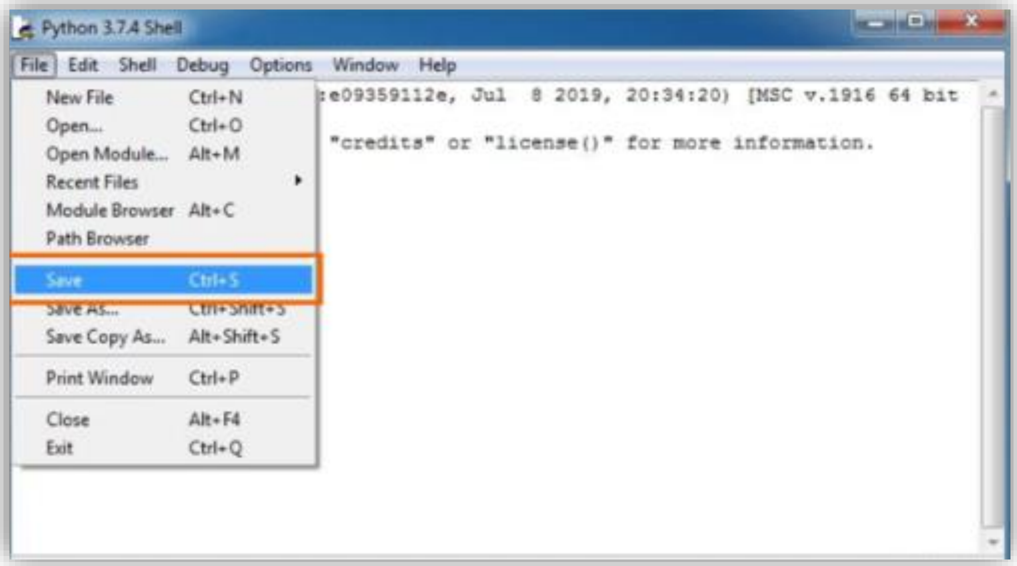


Fig:5.10

Step 3: Click on IDLE (Python 3.7 64-bit) and launch the program

Step 4: To go ahead with working in IDLE you must first save the file. **Click on File > Click on Save**



```
:e09359112e, Jul  8 2019, 20:34:20) [MSC v.1916 64 bit

 "credits" or "license()" for more information.
```

Fig:5.11

Step 5: Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

Step 6: Now for e.g. **enter print**

CHAPTER-6
SYATEM TESTING

6. System 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 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 tests. Each test type addresses a specific testing requirement.

Types Of Tests

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 is invasive. 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 performs accurately to the documented specifications 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 is event driven and is 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 specifically aimed at exposing the problems that arise from the combination of components.

Functional test

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:

Valid Input : identified classes of valid input must be accepted.
Invalid Input : identified classes of invalid input must be rejected.
Functions : identified functions must be exercised.
Output : identified classes of application outputs must be exercised.
Systems/Procedures : interfacing systems or procedures must be invoked.

Organization 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 Test

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 descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is 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 workings, 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, such as specification or requirements document. 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 to outputs without considering how the software works.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

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 the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

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.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

CHAPTER-7
SAMPLE CODE

7. Sample code

```
from tkinter import *
import tkinter
from tkinter import filedialog
import numpy as np
from tkinter.filedialog import askdirectory
from tkinter import simpledialog
from sklearn.model_selection import train_test_split
from tkinter import ttk
from tkinter.filedialog import askopenfilename
import os
import cv2
from keras.models import Sequential
from keras.layers import Input, Dense, LSTM, Embedding
from keras.layers import Dropout, Activation, Bidirectional, GlobalMaxPool1D
from keras.models import Sequential
from sklearn import svm
from sklearn.metrics import accuracy_score
from keras.models import model_from_json
from keras.utils.np_utils import to_categorical
import pickle

main = tkinter.Tk()
main.title("Early Pest Detection From Crop Using Image Processing And Computational Intelligence") #designing main screen
main.geometry("1000x650")

global filename
global X,Y
global X_train, X_test, Y_train, Y_test
global predicted_data
```

```
labels = ['Aphids','Uneffected','Whitefly']
global svm_classifier

def getID(name):
  index = 0
  for i in range(len(labels)):
    if labels[i] == name:
      index = i
      break
  return index

def upload():
  global filename
  filename = filedialog.askdirectory(initialdir = ".")
  text.delete('1.0', END)
  text.insert(END,filename+' Loaded')

def preprocess():
  global X, Y
  global X_train, X_test, Y_train, Y_test
  X = []
  Y = []
  text.delete('1.0', END)
  if os.path.exists('model/X.txt.npy'):
    X = np.load('model/X.txt.npy')
    Y = np.load('model/Y.txt.npy')
    X = X.astype('float32')
    X = X/255
    indices = np.arange(X.shape[0])
    np.random.shuffle(indices)
    X = X[indices]
    Y = Y[indices]
```

```
else:
  for root, dirs, directory in os.walk(filename):
    for j in range(len(directory)):

      name = os.path.basename(root)
      print(name+" "+root+"/"+directory[j])
      if 'Thumbs.db' not in directory[j]:
        img = cv2.imread(root+"/"+directory[j],0)
        img = cv2.resize(img, (28,28))
        im2arr = np.array(img)
        im2arr = im2arr.reshape(28,28)
        X.append(im2arr.ravel())
        Y.append(getID(name))
  X = np.asarray(X)
  Y = np.asarray(Y)
  np.save('model/X.txt',X)
  np.save('model/Y.txt',Y)
  X = np.load('model/X.txt.npy')
  Y = np.load('model/Y.txt.npy')
  X = X.astype('float32')
  X = X/255
  indices = np.arange(X.shape[0])
  np.random.shuffle(indices)
  X = X[indices]
  Y = Y[indices]

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2)
text.insert(END,"Total images found in dataset: "+str(X.shape[0])+"\n\n")
text.insert(END,"Total classes found in dataset: "+str(labels)+"\n\n")
text.insert(END,"Dataset train and test split 80 and 20%\n\n")
text.insert(END,"Training 80% images: "+str(X_train.shape[0])+"\n")
text.insert(END,"Testing 80% images: "+str(X_test.shape[0])+"\n")
```

```
  test = X[3]
  test = test.reshape(28,28)
  test = cv2.resize(test,(200,200))
  cv2.imshow("Process Image",test)
  cv2.waitKey(0)

def runSVM():
  text.delete('1.0', END)
  global svm_classifier
  global X, Y
  global X_train, X_test, Y_train, Y_test

  svm_classifier = svm.SVC()
  svm_classifier.fit(X, Y)
  predict = svm_classifier.predict(X_test)
  svm_acc = accuracy_score(Y_test,predict)*100
  text.insert(END,"SVM Prediction Test Accuracy: "+str(svm_acc))

def runCNN():
  text.delete('1.0', END)
  X_train = np.load('model/X.txt.npy')
  Y_train = np.load('model/Y.txt.npy')
  print(X_train.shape)
  print(Y_train.shape)
  if os.path.exists('model/model.json'):
        with open('model/model.json', "r") as json_file:
            loaded_model_json = json_file.read()
            model = model_from_json(loaded_model_json)
        model.load_weights("model/model_weights.h5")
        model._make_predict_function()
        print(model.summary())
        f = open('model/history.pckl', 'rb')
        data = pickle.load(f)
        f.close()
        acc = data['accuracy']
        accuracy = acc[59] * 100
        text.insert(END,"CNN Model Prediction Accuracy = "+str(accuracy)+"\n\n")
        text.insert(END,"See Black Console to view CNN layers\n")


    else:
        model = Sequential()
        model.add(Convolution2D(32, 3, 3, input_shape = (64, 64, 3), activation = 'relu'))
        model.add(MaxPooling2D(pool_size = (2, 2)))
        model.add(Convolution2D(32, 3, 3, activation = 'relu'))
        model.add(MaxPooling2D(pool_size = (2, 2)))
        model.add(Flatten())
        model.add(Dense(output_dim = 256, activation = 'relu'))
        model.add(Dense(output_dim = 108, activation = 'softmax'))
        print(model.summary())
        model.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy'])
        hist = model.fit(X_train, Y_train, batch_size=16, epochs=60, shuffle=True, verbose=2)
        model.save_weights('model/model_weights.h5')
        model_json = classifier.to_json()
        with open("model/model.json", "w") as json_file:
            json_file.write(model_json)
        f = open('model/history.pckl', 'wb')
        pickle.dump(hist.history, f)
        f.close()
        f = open('model/history.pckl', 'rb')
        data = pickle.load(f)
        f.close()
        acc = data['accuracy']
        accuracy = acc[59] * 100
        text.insert(END,"CNN Model Prediction Accuracy = "+str(accuracy)+"\n\n")
        text.insert(END,"See Black Console to view CNN layers\n")


def checkEffected():
    global svm_classifier
    filename = filedialog.askopenfilename(initialdir="testImages")
    image = cv2.imread(filename,0)
    img = cv2.resize(image, (28,28))

    im2arr = np.array(img)
    im2arr = im2arr.reshape(28,28)
    img = np.asarray(im2arr)
    img = img.astype('float32')
    img = img/255
    temp = []
    temp.append(img.ravel())
    predict = svm_classifier.predict(np.asarray(temp))
    predict = predict[0]
    print(predict)

    img = cv2.imread(filename)
    img = cv2.resize(img, (400,400))
    cv2.putText(img, 'Pest Detected as : '+labels[predict], (10, 25),
cv2.FONT_HERSHEY_SIMPLEX,0.7, (0, 0, 255), 2)
    cv2.imshow('Pest Detected as : '+labels[predict], img)
    cv2.waitKey(0)


def close():
    main.destroy()
```

```
font = ('times', 16, 'bold')
title = Label(main, text='Early Pest Detection From Crop Using Image Processing And Computational Intelligence', justify=LEFT)
title.config(bg='lavender blush', fg='DarkOrchid1')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=100,y=5)
title.pack()

font1 = ('times', 13, 'bold')
uploadButton = Button(main, text="Upload Pest Dataset", command=upload)

uploadButton.place(x=10,y=100)
uploadButton.config(font=font1)

processButton = Button(main, text="Preprocess Dataset", command=preprocess)
processButton.place(x=330,y=100)
processButton.config(font=font1)

svmButton = Button(main, text="Run SVM Algorithm", command=runSVM)
svmButton.place(x=650,y=100)
svmButton.config(font=font1)

svmButton = Button(main, text="Run CNN Algorithm", command=runCNN)
svmButton.place(x=10,y=150)
svmButton.config(font=font1)

testButton = Button(main, text="Check for Effected from Test Image", command=checkEffected)
testButton.place(x=300,y=150)
testButton.config(font=font1)

exitButton = Button(main, text="Exit", command=close)
```

```
exitButton.place(x=650,y=150)
exitButton.config(font=font1)


font1 = ('times', 12, 'bold')
text=Text(main,height=20,width=120)
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=10,y=300)
text.config(font=font1)

main.config(bg='light coral')
main.mainloop()
```

CHAPTER-8
SCREENSHOTS

8. Screenshots

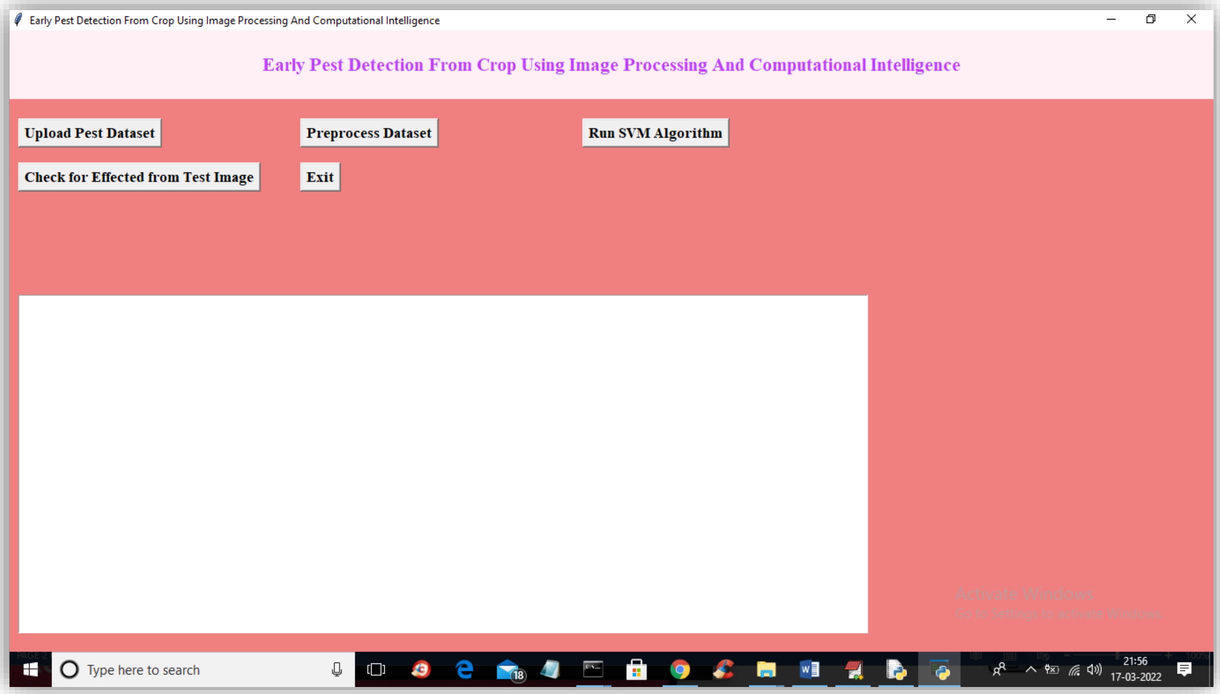
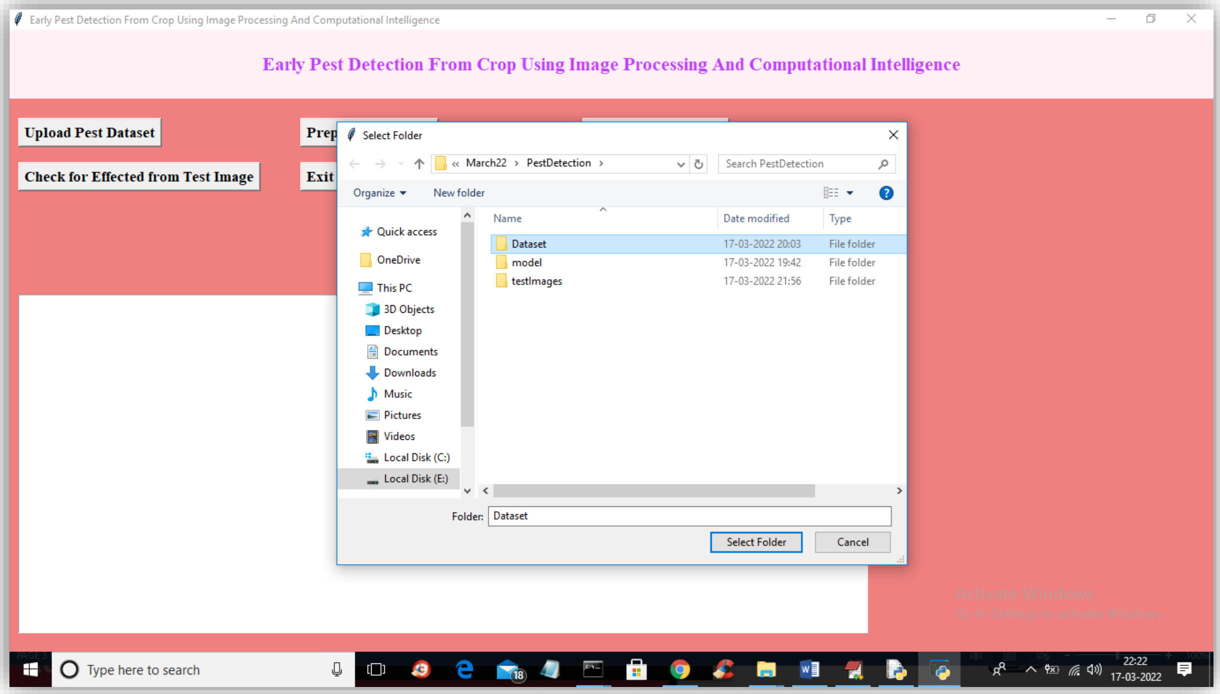

Early Pest Detection From Crop Using Image Processing And Computational Intelligence
Early Pest Detection From Crop Using Image Processing And Computational Intelligence
Upload Pest Dataset
Preprocess Dataset
Run SVM Algorithm
Check for Effected from Test Image
Exit
Type here to search
21:56
17-03-2022

Fig:8.1

In above screen click on 'Upload Pest Dataset' button to upload dataset and to get below screen

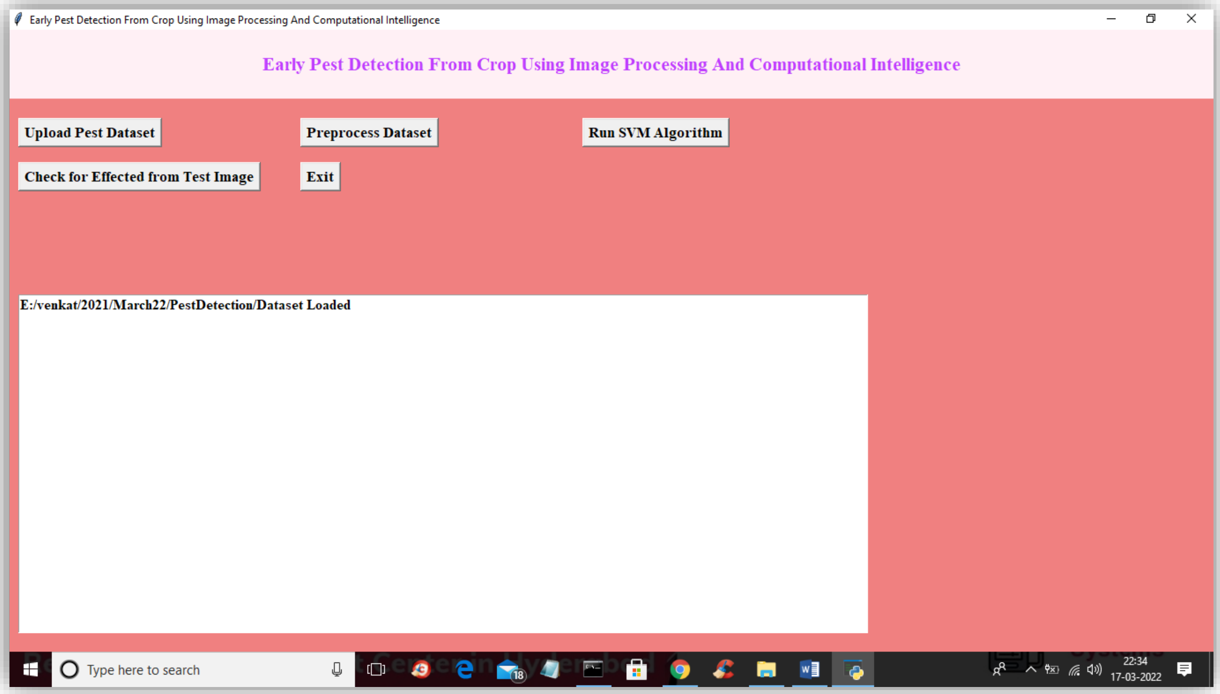


Name	Date modified	Type
Dataset	17-03-2022 20:03	File folder
model	17-03-2022 19:42	File folder
testImages	17-03-2022 21:56	File folder

Folder: Dataset

Fig:8.2

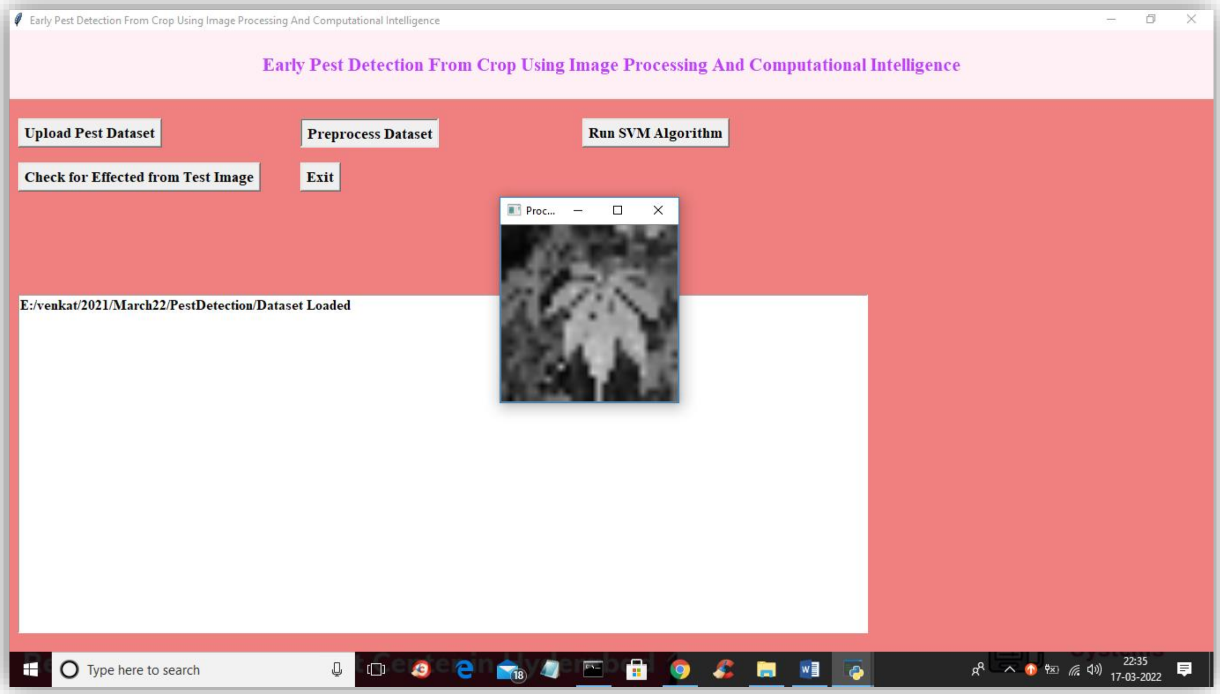
In above screen select and upload 'Dataset' folder and then click on 'Select Folder' button to load dataset and to get below screen.


Early Pest Detection From Crop Using Image Processing And Computational Intelligence
Early Pest Detection From Crop Using Image Processing And Computational Intelligence
Upload Pest Dataset
Preprocess Dataset
Run SVM Algorithm
Check for Effected from Test Image
Exit
Type here to search
22:34
17-03-2022

E:/venkat/2021/March22/PestDetection/Dataset Loaded

Fig:8.3

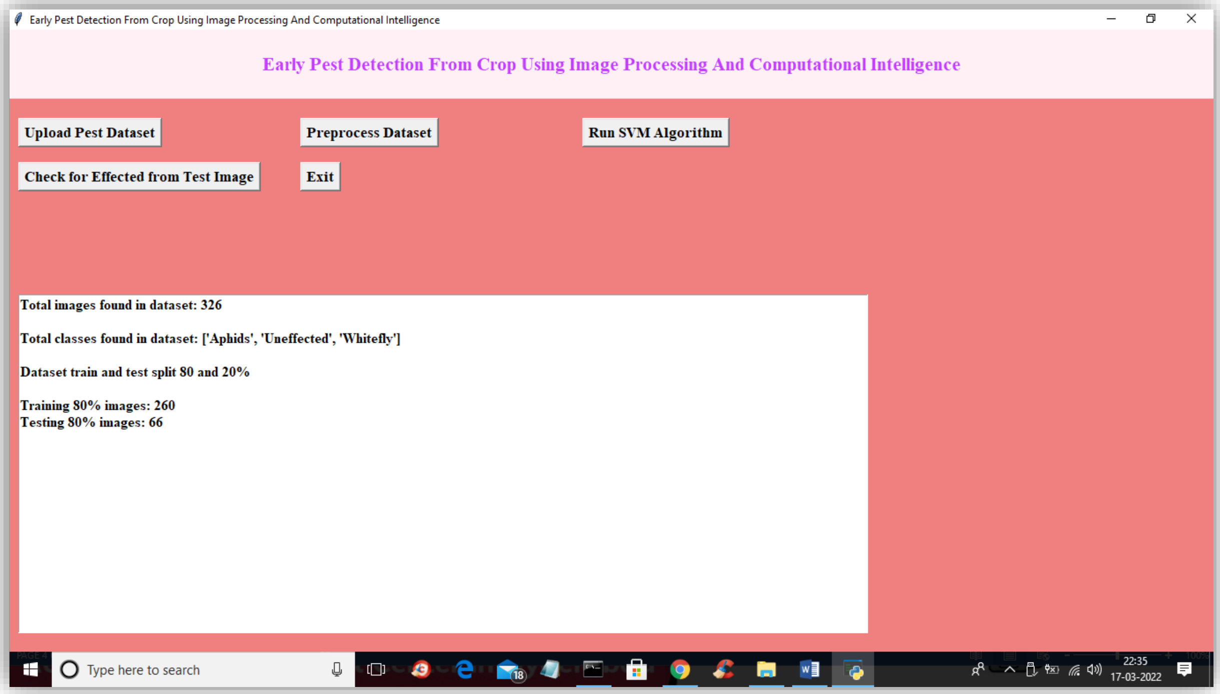
In above screen dataset loaded and now click on 'Preprocess Dataset' button to read and normalize images and then split dataset into train and test part.


Early Pest Detection From Crop Using Image Processing And Computational Intelligence
Early Pest Detection From Crop Using Image Processing And Computational Intelligence
Upload Pest Dataset
Preprocess Dataset
Run SVM Algorithm
Check for Effected from Test Image
Exit
Proc...
Type here to search
22:35
17-03-2022

E:/venkat/2021/March22/PestDetection/Dataset Loaded

Fig:8.4

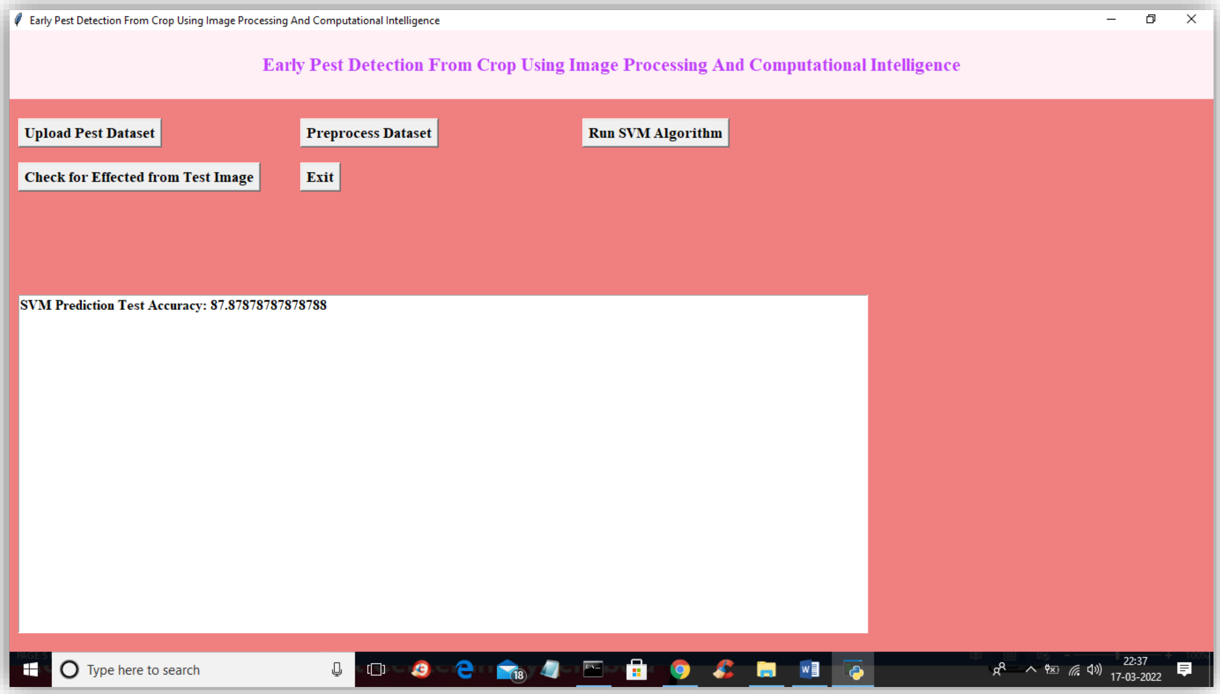
In above screen displaying processed grey image and now close above image to get below screen


Early Pest Detection From Crop Using Image Processing And Computational Intelligence
Early Pest Detection From Crop Using Image Processing And Computational Intelligence
Upload Pest Dataset
Preprocess Dataset
Run SVM Algorithm
Check for Effected from Test Image
Exit
Type here to search
22:35
17-03-2022

```
Total images found in dataset: 326

Total classes found in dataset: ['Aphids', 'Uneffected', 'Whitefly']

Dataset train and test split 80 and 20%

Training 80% images: 260
Testing 80% images: 66
```

Fig:8.5

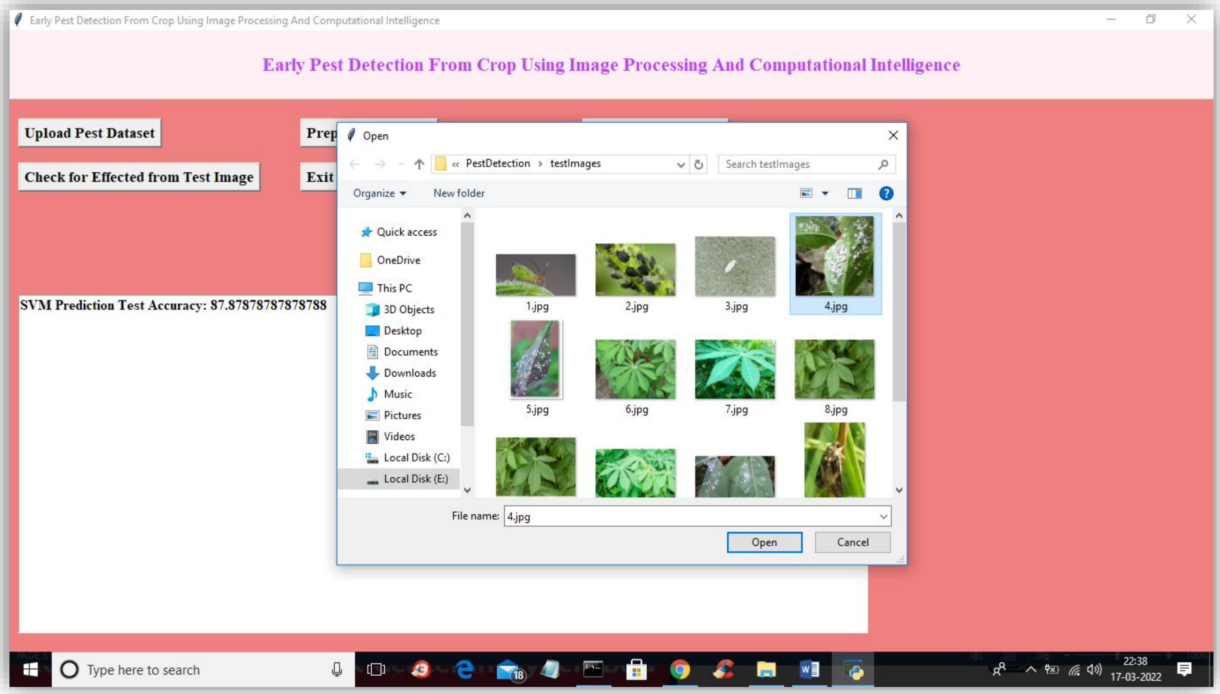
In above screen we can see number of images and classes found in dataset and now click on 'Run SVM Algorithm' button train SVM with processed images and then calculate it's prediction accuracy


Early Pest Detection From Crop Using Image Processing And Computational Intelligence
Early Pest Detection From Crop Using Image Processing And Computational Intelligence
Upload Pest Dataset
Preprocess Dataset
Run SVM Algorithm
Check for Effected from Test Image
Exit
Type here to search
22:37
17-03-2022

```
SVM Prediction Test Accuracy: 87.87878787878788
```

Fig:8.6

In above screen with SVM we got 87% prediction accuracy and now click on 'Check for Effected from Test Image' button to upload test image like below screen



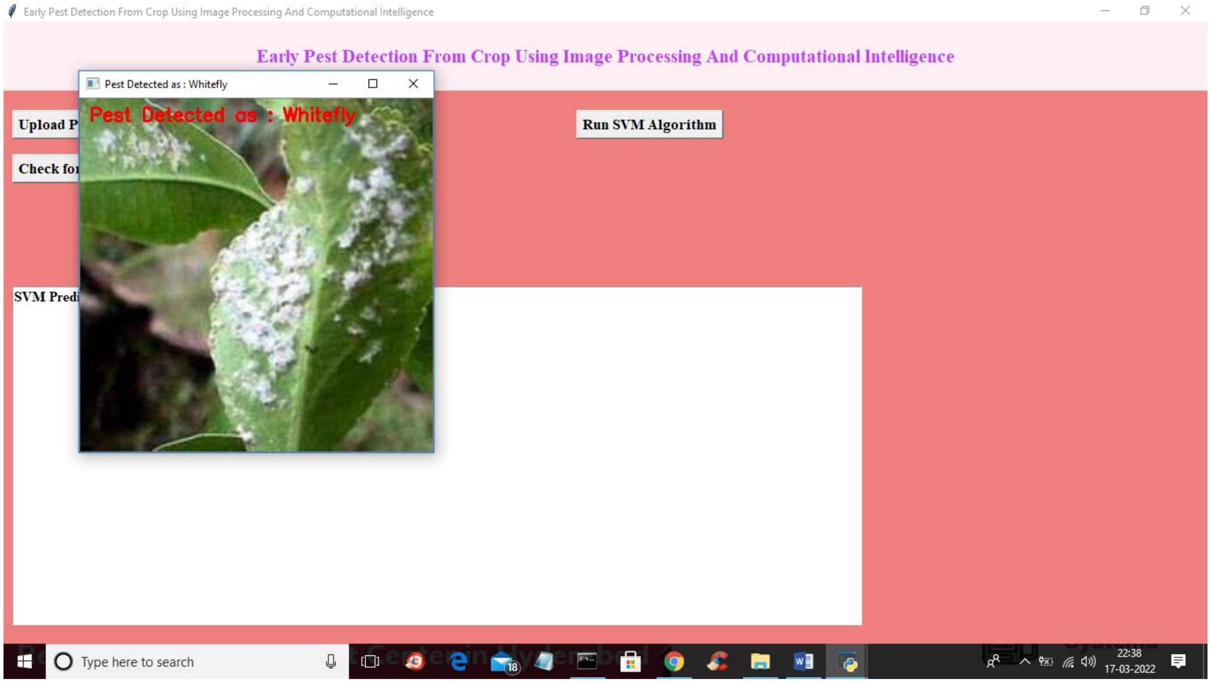
SVM Prediction Test Accuracy: 87.87878787878788

Fig:8.7

In above screen selecting and uploading 4.jpg file and then click on 'Open' button to get below output

CHAPTER-9
RESULTS

9. Results

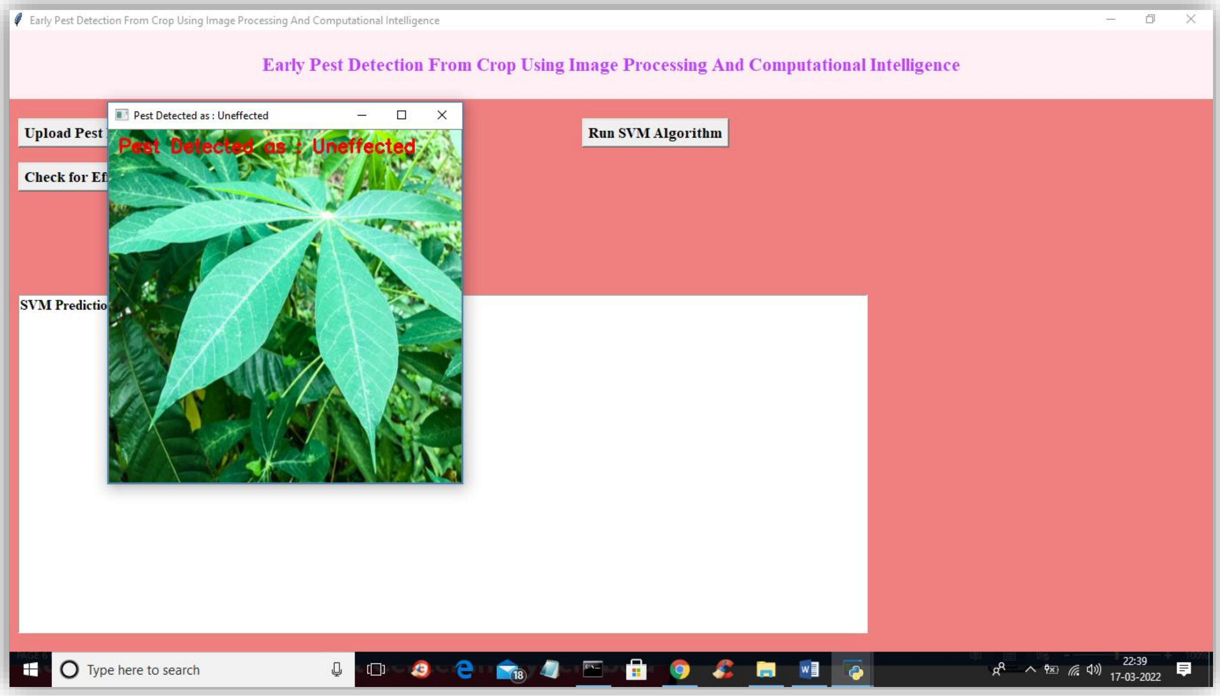

Early Pest Detection From Crop Using Image Processing And Computational Intelligence
Early Pest Detection From Crop Using Image Processing And Computational Intelligence
Upload P
Check fo
Run SVM Algorithm
SVM Predi
Type here to search
22:38
17-03-2022

Pest Detected as : Whitefly

Pest Detected as : Whitefly

Fig:9.1

In above screen in red color text we can see SVM predicted/classified uploaded image as ‘whitefly’ and similarly you can upload and test other images


Early Pest Detection From Crop Using Image Processing And Computational Intelligence
Early Pest Detection From Crop Using Image Processing And Computational Intelligence
Upload Pest
Check for Ef
Run SVM Algorithm
SVM Predictio
Type here to search
22:39
17-03-2022

Pest Detected as : Uneffected

Pest Detected as : Uneffected

Fig:9.2

In above screen uploaded image is predicted as ‘Uneffected’ as it not contains any pest.

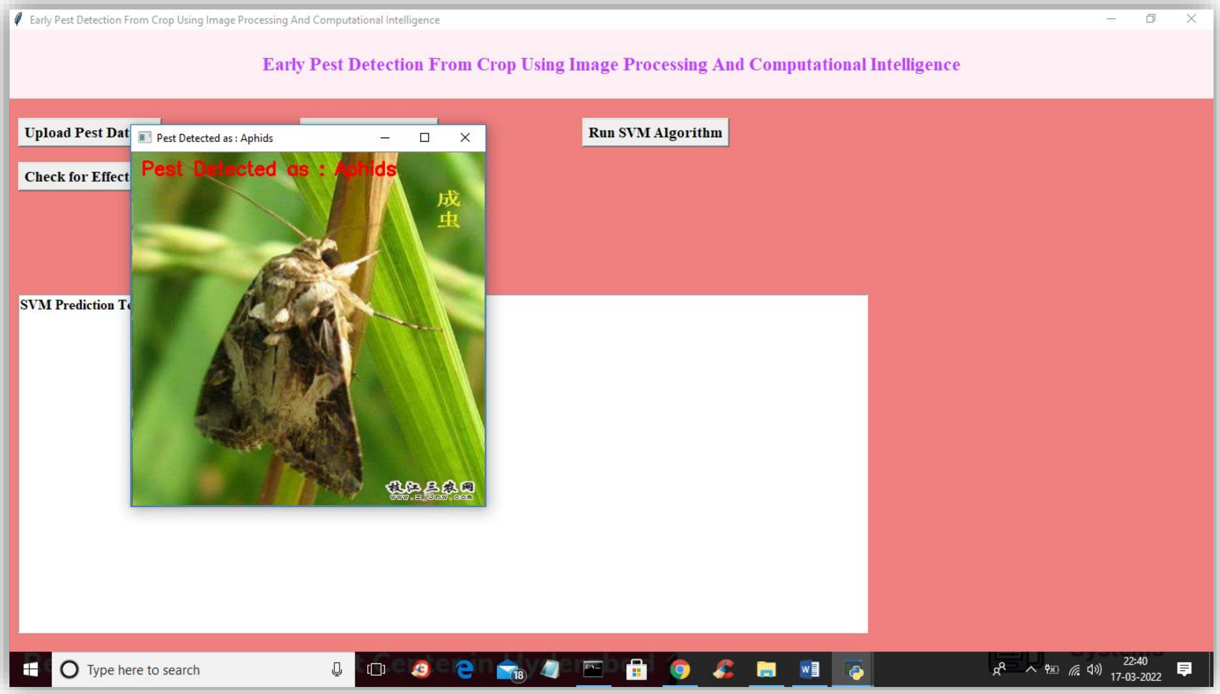

Early Pest Detection From Crop Using Image Processing And Computational Intelligence
Early Pest Detection From Crop Using Image Processing And Computational Intelligence
Upload Pest Dat
Check for Effect
Run SVM Algorithm
SVM Prediction Te
Pest Detected as : Aphids
Pest Detected as : Aphids
成
虫
Type here to search
22:40
17-03-2022

Fig:9.3

In above screen uploaded image is classified as 'Aphids'

CHAPTER-10
CONCLUSION & FUTURE SCOPE

10. Conclusion & Future scope

Conclusion

Image processing technique plays an important role in the detection of the pests. Our first objective is to detect whiteflies, aphids, and thrips on greenhouse crops. We propose a novel approach for early detection of pests.

To detect objects, we use pan tilt camera with zoom. So, without disturbing the pests we can take the image. It illustrates the collaboration of complementary disciplines and techniques, which led to an automated, robust, and versatile system.

The prototype system proved reliable for rapid detection of pests. It is rather simple to use and exhibits the same performance level as a classical manual approach. Our goal is to detect the pests as early as possible and reduce the use of pesticides.

Future scope

Disease detection, quantification, classification and prediction aresuch challenging domains as they contain many varying parameters. Due to its vast, unpredictable nature inclusion of the latest machine learning and big data techniques will be a major improvement and an obvious evolution. discussed the implementation of big data in farming practices which further improves the predictiveness of external environmental factors in farming and many others. There are many challenges still in the field of plant disease diagnosis using image processing and computer vision. In the recent review papers, we found out there are many potentials areas such as work on 3D images are still notso prevalent. Talking about deep learning, the predictive and probabilistic model generation from the already existing data is one of the significant advantages of it, and hence it is quite efficient regarding plant growth and disease prediction, identification, classification and quantification. Furthermore, multiple kernel support vector regression and fusion of various techniques, such as super pixel along with | PHOG, can be explored. Better optimization and segmentation techniques with newer evolutionary methods can be examined. Use of portable development boards such as Raspberry Pi, Beagle Bone, Intel Galileo are still not so prevalent which can help in portability, cost-effectiveness and robustness of the system. The inclusion of IoT related framework will also boast the work and enhance the performance results.

CHAPTER-11
REFERENCES

11. References

[1]. Martin, V. Thonnat, “A Learning Approach For Adaptive Image Segmentation.” In Proceedings of IEEE Trans.Computers and Electronics in Agriculture.2008

[2]. Vincent Martin and Sabine Moisan, “Early Pest Detection in Greenhouses”. INRIA Sophia Antipolis Mediterranean, PULSAR team 2004 route des Lucioles, BP93

[3]. Jayamala K. Patil1, Raj Kumar, “Advances In Image Processing For Detection Of Plant Diseases” Journal Of Advanced Bioinformatics Applications and Research ISSN 0976-2604 Vol 2, Issue 2, June-2011, pp 135-141

[4]. Ikhlef Bechar and Sabine Moisan, “On-line counting of pests in a greenhouse using computer vision”. published in "VAIB 2010 - Visual Observation and Analysis of Animal and Insect Behavior (2010)"

[5]. Paul Boissarda, Vincent Martin, “A cognitive vision approach to early pest detection in greenhouse crops”. computers and electronics in agriculture 6 2 (2 0 0 8) 81–93

[6]. B. Cunha. “Application of Image Processing in Characterisation of Plants.” IEEE Conference on Industrial Electronics.2003.