Fertilizer Recommendation System for Disease Prediction

USING ARTIFICIAL INTELLIGENCE

A Project report submitted in partial fulfilment of 7th semester in degree of

BACHELOR OF ENGINEERING
IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

Team ID: PNT2022TMID46085

Gokulalakshmi R	814819104011
Bhavani N	814819104007
Sneha R	814819104027
Sneka V	814819104028
Madhubala T	814819104015

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING UNIVERSITY COLLEGE OF ENGINEERING, ARIYALUR

ANNA UNIVERSITY: CHENNAI 600025

NOV-2022

UNIVERSITY COLLEGE OF ENGINEERING,ARIYALUR
(A Constituent College of Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **"FERTILIZER RECOMMENDATION SYSTEM FOR DISEASE PREDICTION"** is the bonafide record work done by **R GOKULALAKSHMI** (**814819104011**),**N BHAVANI** (**814819104007**), **R SNEHA** (**814819104027**),**V SNEKA** (**814819104028**) and **T MADHUBALA(814819104015)** for **"HX 8001 PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP"** in **VII** semester of **B.E.,** degree course **in ComputerScience and Engineering** branch during the academic year of 2022 - 2023.

Staff-In charge
Mrs.N.Shanmugapriya

Evaluator
Dr.R.Arunprakash

Head of the Department
Dr.S.Jayanthi

FERTILIZERS RECOMMENDATION SYSTEM FOR DISEASE PREDICTION

ABSTRACT

Agriculture serves as a means of supplying food to a population that is always expanding, as well as a significant source of energy and a means of combating global warming. Plant diseases are very important because they can have a negative impact on the quality and quantity of crops produced in agriculture. Early detection of plant diseases is crucial for their treatment and management. Typically, illnesses are identified using the naked eye technique. Experts who can recognise variations in leaf colour are involved in this process. This method requires a lot of work, takes a while, and is not appropriate for fields with a lot of space.The same ailment is frequently classified differently by various experts. Costly specialist monitoring is required for this procedure, which makes it pricey. Plant diseases can drive up the cost of agricultural production and, if left untreated at an early stage, could spell complete financial ruin for a producer. In order to stop the spread of a plant disease at a low cost and save the majority of the production, farmers must keep an eye on their crops and recognise the first symptoms. It may be expensive to hire experienced agriculturists, particularly in remote, isolated geographic areas.Various experts regularly assign multiple classifications to the same illness. This operation is costly because it calls for pricy professional supervision. Plant diseases can increase the cost of agricultural production and, if not promptly treated, could result in a producer's total financial ruin. Farmers must keep an eye on their crops and be able to spot the first symptoms in order to stop the spread of a plant disease at a low cost and save the majority of the production. Agriculturists with experience may be expensive to hire, especially in isolated, distant locations.

INTRODUCTION

Digital image processing is the use of computer algorithms to perform image processing on digital images. As a subcategory or field of digital signal processing, digital image processing has many advantages over analog image processing. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and signal distortion during processing. Since images are defined over two dimensions (perhaps more) digital image processing may be modeled in the form of multidimensional systems.

Many of the techniques of digital image processing, or digital picture processing as it often was called, were developed in the 1960. The cost of processing was fairly high, however, with the computing equipment of that era. That changed in the 1970s, when digital image processing proliferated as cheaper computers and dedicated hardware became available. Images then could be processed in real time, for some dedicated problems such as television standards conversion. As general-purpose computers became faster, they started to take over the role of dedicated hardware for all but the most specialized and computer-intensive operations. With the fast computers and signal processors available in the 2000s, digital image processing has become the most common form of image processing and generally, is used because it is not only the most versatile method, but also the cheapest.

Digital Processing techniques help in manipulation of the digital images by using computers. As raw data from imaging sensors from satellite platform contains deficiencies. To get over such flaws and to get originality of information, it has to undergo various phases of processing. The three general phases that all types of data have to undergo while using digital technique are Pre- processing, enhancement and display, information extraction.

Segmentation procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually. Segmentation is a classifier which helps to fragment each character from a word present in a given image or page. The objective of the segmentation is to extract each character from the text present in the image. After performing Segmentation, the characters of the string will be separated and it will be used for further processing. Different character segmentation techniques has been proposed until like, Dissection Techniques, Recognition Based Hidden Markov Models and Non-Markov Approaches, Holistic Strategies. By dissection is meant

the decomposition of the image into a sequence of sub images using general features. The structure consists of a set of states plus transition probabilities between states.

1.1 OBJECTIVE

It can be challenging to diagnose some diseases with the naked eye when they are present but not readily apparent. And once it becomes evident, it will be too late to diagnose the illness and render assistance. A microscope was once used to find the disease, but it is now challenging to watch every leaf and plant. The quick and efficient method is therefore a remote sensing technique. The early identification of disease symptoms is made possible by the detection and recognition of plant diseases using machine learning. For the purpose of diagnosing plant diseases, plant pathologists can examine digital photographs utilising digital image processing.For agricultural applications, such as the identification of leaf diseases, fruit illnesses, etc., computer processing systems are developed. All of these methods include the collection of digital images using a digital camera and the application of image processing techniques to these images in order to extract meaningful information required for additional research. Digital image processing is used to implement the process that will take an image as input, process it, and then provide the desired or anticipated result. Farmers can benefit from the use of computer vision and image processing technology in various facets of agricultural work.

1.2 SCOPE

Data mining is a computational process that combines techniques from machine learning, statistics, and database systems to find patterns in big data sets. This area of computer science is interdisciplinary. The main objective of data mining is to take information from a data set and organise it so that it can be used in other ways. The analytical stage of the "knowledge discovery in databases" process is known as data mining. The procedure used to look for patterns in huge data sets is called data mining, which is a field at the intersection of computer science and statistics. Data mining is the analytical stage of the "Knowledge Discovery in Databases" process, or KDD. It makes use of techniques that combine machine learning, statistics, systems, and artificial intelligence.

The actual data mining task is the semi-automatic or automatic analysis of large quantities of data to extract previously unknown, interesting patterns such as groups of data records (cluster analysis), unusual records (anomaly detection), and dependencies (association rule mining, sequential pattern mining). This usually involves using database techniques such as spatial indices. These patterns can therefore be viewed as a sort of summary of the input data and can be applied to further analysis, as well as to machine learning and predictive analytics, for example. A decision support system may leverage the many groups that were discovered during the data mining process, for instance, to produce predictions that are more accurate.

1.3 PURPOSE

They forecast plant disease and recommend fertiliser for the damaged plants. This frequently involves a range of methods for assessing the qualities of the herbs that largely influence the plants. These complex systems that contain a large amount of datasets are forecasted using the neural network. By using artificial intelligence, complex manual systems' working models can be made simpler and more precise.

1.4 FEASIBILITY STUDY

Objectives of Feasibility Study

- To explain present situation of the automation.
- To find out if a system development project can be done is possible.
- To find out whether the final product will benefit end user.
- To suggest the possible alternative solutions.

SYSTEM STUDY

FEASIBILITY STUDY

The purpose of this chapter is to introduce the reader to feasibility studies, project appraisal, and investment analysis. Feasibility studies are an example of systems analysis. A system is a description of the relationships between the inputs of labour, machinery, materials and management procedures, both within an organisation and between an organisation and the outside world.

During the planning and execution stages of an audit, it's important to have a clear understanding of what the objectives of the audit include. Companies should strive to align their business objectives with the objectives of the audit. This will ensure that time and resources spent will help achieve a strong internal control environment and lower the risk of a qualified opinion.

Objectives of Feasibility Study

- To explain present situation of the automation.
- To find out if a system development project can be done is possible.
- To find out whether the final product will benefit end user.
- To suggest the possible alternative solutions.

Technical Feasibility

Technical Feasibility assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team is capable of converting the ideas into working systems.In technical feasibility the following issues are taken into consideration.

- Whether the required technology is available or not
- Whether the required resources are available - Manpower- programmers, testers & debuggers, Software and hardware

Once the technical feasibility is established, it is important to consider the monetary factors also. Since it might happen that developing a particular system may be technically possible but it may require huge investments and benefits may be less. For evaluating this, economic feasibility of the proposed system is carried out.

Economical Feasibility

Economic feasibility analysis is the most commonly used method for determining the efficiency of a new project. It is also known as cost analysis. It helps in identifying profit against investment expected from a project. Cost and time are the most essential factors involved in this field of study.For any system if the expected benefits equal or exceed the expected costs, the system can be judged to be economically feasible. In economic feasibility, cost benefit analysis is done in which expected costs and benefits are evaluated. Economic analysis is used for evaluating the effectiveness of the proposed system.

Operational Feasibility

Operational Feasibility is depend on human resources available for the project and involves projecting whether the system will be used if it is developed and implemented. Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements analysis phase of system development. Operational feasibility reviews the willingness of the organization to support the proposed system. This is probably the most difficult of the feasibilities to gauge. In order to determine this feasibility, it is important to understand the management commitment to the proposed project. If the request was initiated by management, it is likely that there is management support and the system will be accepted and used. However it is also important that the employee base will be accepting of the change.

Schedule Feasibility

In this type of feasibility, the skills required for properly applying the new technology with training in minimum time and the time duration can be checked out to implement or overrun the new project within minimum time. Schedule feasibility ensures that a project can be

completed before the project or technology becomes obsolete or unnecessary. Schedule feasibility can be calculated using research period.

SYSTEM ANALYSIS

System analysis is a procedure that separates the functions of the system based on the characteristics of the problem domain and the needs of the users.

2.1 EXISTING SYSTEM

Leaves are the most obvious and widespread choice for tree species recognition, even though the botanical classification was not built upon their properties. They can be found almost all year long, are easy to photograph, and their shapes present well studied specificities that make the identification, if not trivial, possible. Our goal with the Folia application is then to build a system for leaf shape analysis that processes, unlike what has been done to date, pictures in a natural environment. With the aim of being an educational tool, it relies on high-level geometric criteria inspired by those used by botanists, that make a semantic interpretation possible, to classify a leaf into a list of species. Digital image processing will improve the quality of the image by removing noise & other unwanted pixels and obtain more information from image. Image segmentation is a mid-level processing technique used to analyze the image and can be used to classify or cluster an image into several disjoint parts by grouping the pixels to form a region of homogeneity based on the pixel characteristics like gray level, color, texture, intensity and other features. The main purpose of the segmentation process is to get more information about the image, the region we are interested in and to clearly differentiate the object and the background in an image. The criteria for segmenting the image is very hard to decide as it varies from image to image and also varies significantly on the modal quality of image. In some cases interactive methods can be laborious and time consuming and in some cases manual interaction to segment the image may be error-prone while the fully automated approach can give error output.

2.1.1 LIMITATIONS OF EXISTING SYSTEM:

- Suffer in the local minima problems.
- Dimensionality is high to produce large number of irrelevant features.
- User defined segmentation can be done.
- Does not recommend the fertilizers to leaves diseases.
- Manual approach is used

2.2 PROPOSED SYSTEM

Even when considering trees only, leaves show an impressively wide variety in shapes. It is however necessary to come up with a representation of what a leaf is, that is accurate enough to be fitted to basically any kind of leaf. The general shape of a leaf is a key component of the process of identifying a leaf. Botanists have a whole set of terms describing either the shape of a simple leaf, of the lobes of a palmate leaf, or of the leaflets of a compound leaf. Here present a study on segmentation of leaf images restricted to semi-controlled conditions, in which leaves are photographed against a solid light-colored background. Such images can be used in practice for plant species identification, by analyzing the distinctive shapes of the leaves. The most important of these are: the variety of leaf shapes, inevitable presence of shadows and specularities, and the time constraints required by interactive species identification applications. The identification of species is the first and essential key to understand the plant environment. In this project introduce a method designed to deal with the obstacles raised by such complex images, for simple and lobed tree leaves. A first segmentation step based on a light polygonal leaf model is first performed, and later used to guide the evolution of an active contour. Combining global shape descriptors given by the polygonal model with local curvature-based features, the leaves are then classified over leaf datasets. In this project we introduce a method designed to deal with the obstacles raised by such complex images, for simple and lobed tree leaves. A first segmentation step based on graph cut approach is first performed, and later used to guide the evolution of leaf boundaries. And implement classification algorithm to classify the diseases and recommend the fertilizers to affected leaves.

2.1.1 ADVANTAGE OF PURPOSED SYSTEM:

- Segmentation can be done easily to spilt the tree parts.
- Classify the affected parts in leaves.
- Eliminate redundant features of images.
- Provide improved accuracy rate.

2.2 REQUIREMENT SPECIFICATION

The technical specification requirement for the software products is the requirement specification. It lists the functional, performance, and security requirements for a specific software system. Additionally, usage scenarios from a user, operational, and administrative standpoint are provided in the requirements. A thorough overview of the software project is what the software requirements specification is meant to do. The target audience is given a description of the project's parameters, goals, user interface, hardware, and software needs.

2.2.1 HARDWARE SPECIFICATION

- Processor : Dual core processor 2.6.0 GHZ
- RAM : 2GB
- Hard disk : 160 GB
- Compact Disk : 650 Mb
- Keyboard : Standard keyboard
- Monitor : 15 inch color monitor

2.2.2 SOFTWARE SPECIFICATION

- Operating system : Windows OS
- Front End : C#.NET
- Back End : SQL SERVER
- Application : Windows Application
- Tool : Visual Studio 2010

2.4 OVERVIEW OF SOFTWARE

Special Features of C# . Net

Dot Net Overview Visual Studio .NET

Visual Studio .Net is the fast application improvement device for BASIC. Visual Studio .Net offers complete mix with C#.NET and empowers to move and customize server controls and outline Web Forms as they ought to show up when client sees them. A percentage of alternate points of interest of making BASIC applications in Visual Studio .Net are

- Visual Studio .Net is a Rapid Application (RAD) apparatus. Rather than adding every control to the Web Form automatically, it serves to include these controls by utilizing tool stash, sparing programming endeavors.

- Visual Studio .Net backings custom and composite controls. Can make custom controls that embody a typical usefulness that may need to use in various applications.

- Visual Studio .Net makes a glorious showing of rearranging the creation and utilization of Web Services. Mush of the software engineer neighborly stuff (making all the XML-based reports) happens consequently, without much exertion on the developer's side.

- A characteristic based writing computer program is an effective idea that empowers Visual Studio .Net to mechanize a considerable measure of software engineer unpleasant assignments.

.NET Programming Dialects:

The .NET Framework gives an arrangement of instruments that assistance to assemble code that works with the .NET Framework, Microsoft gives an arrangement of dialects that are as of now .NET perfect. Fundamental is one of those dialects.

C#.NET Environment:

Dynamic Server Pages were discharged by Microsoft to empower the formation of element pages taking into account client information and cooperation with a Web website. C#.NET enhances the first ASP by giving code-behind. With C#.NET and code-behind, the code and HTML can be isolated.

C#.NET Web administrations are XML-construct benefits that are presented with respect to the Internet that can be gotten to by other Web administrations and Web administration customers.

C#.NET:

C#.NET is more than the following form of Active Server Pages (ASP); it is a brought together Web advancement stage that gives the administrations important to designers to fabricate undertaking class Web applications. While C#.NET is to a great extent sentence structure perfect with ASP, it likewise gives another programming model and foundation for more secure, versatile, and stable applications.

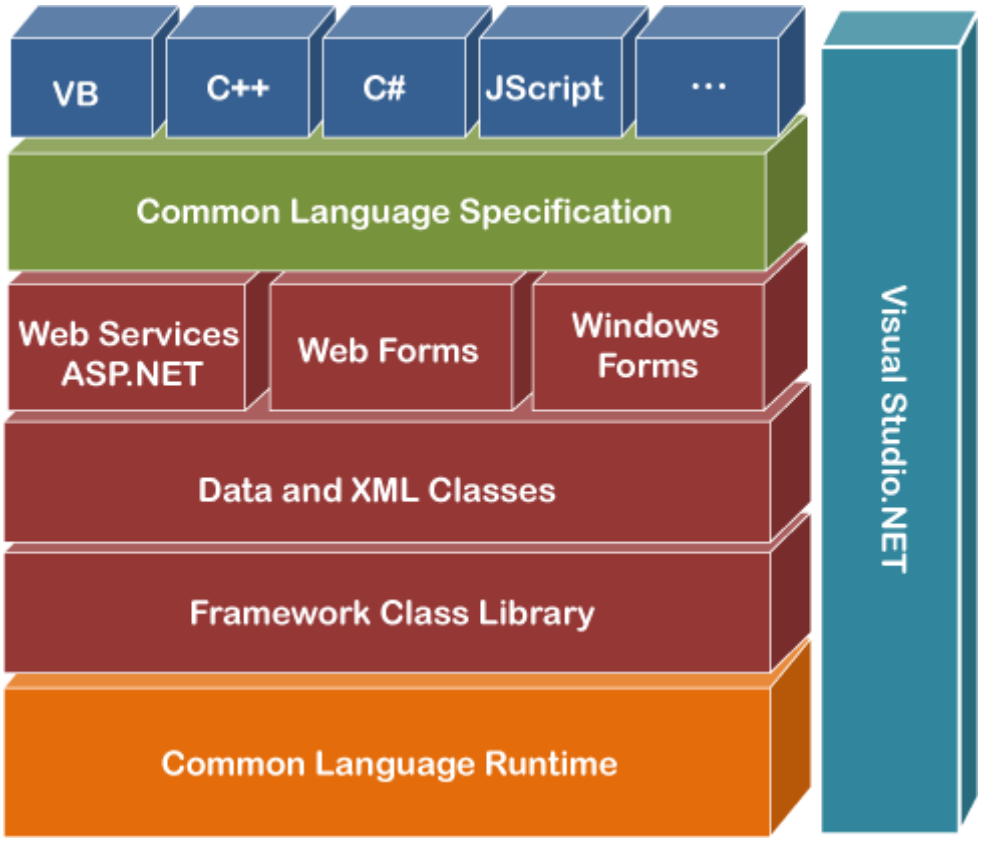
C#.NET is an assembled, .NET-based environment; you can creator applications in any .NET perfect dialect, including VisualBasic.NET, BASIC, and JScript.NET. Furthermore, the whole .NET Framework is accessible to any C#.NET application. Engineers can undoubtedly get to the regale of these advances, which incorporate oversaw normal dialect runtime environment, sort wellbeing, legacy, et cetera.

C#.NET has been intended to work consistently with WYSIWYG HTML editors and other programming instruments, including Microsoft Visual Studio .NET. Does this make Web improvement simpler, as well as gives every one of the advantages that these apparatuses bring to the table, including a GUI that designers can use to drop server controls onto a Web page and completely coordinated investigating backing. Engineers can browse the accompanying two elements when making a " C#.NET application, Web Forms and Web administrations, or consolidate these in any capacity they see fit.

- Web Forms permits you to assemble intense structures based Web pages. At the point when building these pages, you can utilize C#.NET server controls to make normal Ul components, and system them for basic assignments. These controls permit you to quickly assemble a Web Form out of reusable implicit or custom segments, rearranging the code of a page.

- An XML Web administration gives the intends to get to server usefulness remotely

C#.NET Architecture:



Features:

- **Intuitive C++ Based Language**

Utilize a dialect displayed on C++ linguistic structure, instantly commonplace to C++ and Java designers, and also natural new dialect builds that incredibly streamline advancement errands

- **Reliable Interoperability**

Utilize code to call local Windows APIs, use pre-constructed COM parts, and influence existing ActiveX controls to flawlessly coordinate existing applications and segments.

- **Advanced, Component-Oriented Language**

Exploit inborn backing for properties, indexers, delegates, single and multidimensional clusters, propelled legacy, traits, forming, and XML remarks.

- **Capable Debugging and Testing Tools**

C# .NET incorporates a capable remote and multi-dialect debugger, empowering engineers to test applications and fabricate solid multi-level arrangements that compass process limits and are composed in different programming dialects.

Net Framework Class Library :

Addition experienced and capable, constructed in usefulness, including a rich arrangement of accumulation classes, systems administration bolster, multithreading bolster, string and customary expression classes, and wide backing for XML, XML patterns, XML namespaces, XSLT, XPath, and SOAP.

Powerful Web Development Environment:

Make Web-based arrangements in C# utilizing the mutual Web Forms Designer and XML Designer. Engineers can likewise utilize IntelliSense elements and label finish or pick the WYSIWYG manager for move and customize creating to construct intelligent Web applications.

. NET Framework:

Microsoft planned VB from the beginning to exploit its new .NET Framework. The .NET Framework is comprised of four sections, the Common Language Runtime, an arrangement of class libraries, an arrangement of programming dialects, and the C#.NET environment. The .NET Framework was composed on account of three objectives. In the first place, it was planned to make Windows applications considerably more solid, while likewise furnishing an application with more prominent level of security.

Second, it was proposed to improve the advancement of Web applications and administrations that work in the conventional sense, as well as on cell phones too. Finally, the structure was intended to give a solitary arrangement of libraries that would work with various dialects. The .NET Framework is the base for the new Microsoft .NET Platform. Furthermore, it is a typical situation for building, conveying, and running Web applications and Web Services. The .NET Framework contains a typical dialect runtime and basic class libraries - like ADO .NET, ASP .NET and Windows Forms - to give propelled standard administrations that can be coordinated into a mixed bag of PC frameworks. The .NET Framework gives a component rich application environment, streamlined improvement and

simple mix between various diverse advancement dialects. The .NET Framework is dialect nonpartisan. At present it bolsters C++, C#, Visual Basic, and Jscript. Microsoft's Visual Studio.NET is a typical advancement environment for the new .NET Framework.

Coordinating with IIS

IIS is the web server is utilized here. IIS 5.0 or above is key for the C#.NET for the earth. This arrival of C#.NET uses IIS 5.0 as the priKim host environment. IIS dependably accept that an arrangement of accreditations maps to a Windows NT record and uses them to verify a client. There are three various types of validation accessible in IIS 5.0: BASIC, DIGEST, and INTEGRATED WINDOWS Authentication (NTLM or Kerberos). You can choose the kind of verification to use in the IIS regulatory administrations.

On the off chance that you ask for a URL containing an C#.NET application, the solicitation and confirmation data are given off to the application. C#.NET gives the two extra sorts of verification depicted in the accompanying table.

Web Service:

Web administrations are ostensibly the most energizing and improve elements of Microsoft's. NET activity and they are liable to significantly influence the way business collaborate utilizing PC application. Rundown of conceivable Web administrations is as changes as the rundown of conceivable business opportunities. Web administration would normally perform a center business administration, for example, client confirmation, Visa approval, valuing a derivates security, submitting a buy request for a stock or estimating a same-day shipment.

A web administration is a part that performs a capacity or administration. A segment is a bit of programming that has a very much characterized interface, shrouded internals, and the ability of being found. By "found" implies that you can figure out what the part' manages without expecting to see the code inside of it. A segment is like a strategy since we can call it with contentions that fit an arrangement of parameters, and it has the ability of returning results.

A web administration might likewise return data to the guest. This administration dwells some place on the Web and can be gotten to from different areas on the Web. For this administration to be called, there are various components that must be set up. To start with,

the guest must' know how to call the administration. Second, the call must be made over the Web. At long last, the “web administration must know how to react”.

6.2 About SQL Server

SQL Server 8.0:

Social database frameworks are the most critical database frameworks utilized as a part of the product business today. A standout amongst the most remarkable frameworks is Microsoft SQL Server. SQL Server is a database administration framework created and showcased by Microsoft. It runs solely under Windows NT and Windows 95/98.

- **The most critical parts of SQL Server 8 are:**

- SQL Server is anything but difficult to utilize.
- SQL Server scales from a portable tablet to symmetric multiprocessor frameworks.
- SQL Server gives information warehousing elements that as of recently have just been accessible in Oracle and other more costly DBMSs.

A database framework is a general gathering of distinctive database programming segments and databases containing the parts viz. Database application projects, Front-End segments, Database administration frameworks, and Databases.

- **A database framework must give the accompanying elements:**

- A mixture of client interfaces
- Physical information autonomy
- Logical information autonomy
- Query advancement
- Data honesty
- Concurrency control
- Backup and recuperation
- Security and approval

SQL Server is a Relational Database Management System. The SQL Server social dialect is called Transact-SQL.SQL is resource arranged dialect. This implies that SQL can inquiry numerous lines from one or more tables utilizing only one announcement. This component permits the utilization of this dialect at a coherently larger amount than procedural dialects. Another vital property of SQL is its non-procedurally. SQL contains two sub dialects DDL and DML.

SQL Server functions as a characteristic augmentation of Windows NT and windows 95/98.SQL Server is generally simple to oversee through the utilization of a graphical registering environment for each undertaking of framework and database organization. SQL Server uses administrations of Windows NT to offer new or expanded database capacities, for example, sending and accepting messages and overseeing login security.

The SQL Server chairman's essential device for connecting with the framework is Enterprise Manager. The Enterprise Manager has two primary purposes: Administration of the database server and Management of database items.

- SQL Server Query Analyzer gives a graphical presentation of the execution arrangement of a question and a programmed segment that recommends which list ought to be utilized for a chose inquiry. This intelligent segment of SQL Server performs the assignments like:

- Generating and executing Transact-SQL explanations
- Putting away the produced Transact-SQL explanations in a document
- Analyzing execution gets ready for produced inquiries
- Graphically representing the execution arrangement for a chose question.

A put away method is an exceptional sort of clump written in Transact-SQL utilizing the SQL dialect and SQL augmentations. It is saved money on the database server to enhance the execution and consistency of monotonous undertakings. SQL Server backings put away methods and framework techniques. Put away techniques can be utilized for the accompanying purposes: to control access approval, to make a review trial of exercises in database tables, to discrete information definition & information control articulations concerning a database & every single comparing application.

The database article perspective can be utilized for:

- Restricting the utilization of specific sections and lines of tables - that is to control access to a specific piece of one or more tables,

- To shroud the points of interest of confounded inquiries, to limit embedded & redesigned qualities to certain extents.

The Query Optimizer is the piece of SQL Server that chooses how to best perform a question. It creates a few inquiry execution gets ready for the given question & chooses the arrangement with the most minimal expense.

SQL Server can work in one of two security modes:

- Windows NT

- Mixed

Windows NT security mode solely utilizes Windows NT client records to sign into the SQL Server framework. Blended mode permits clients to associate with SQL Server utilizing the Windows NT security framework or the SQL Server framework. Moreover it gives three security offices to controlling access to database objects:

- Transact-SQL explanations GRANT, DENY, and REVOKE.

- Views.

- Stored methodology

A Windows NT client record or a SQL server login name permits a client to sign into the SQL server framework. A client who hence needs to get to a database of the framework needs a database client record to work in the DB. In this manner clients must have a DB client represent each DB they need to utilize. In the event that there is no such record the client may be permitted to work in the DB under the visitor account."

Put away methods can likewise be utilized to limit information access. The confinement of information access utilizing put away methodology is based upon the property that the consent to execute a put away' strategy is free of any authorization for DB objects that are referenced by the put away system.

SQL server gives an instrument called a trigger for upholding procedural respectability requirements.

A DBMS Handles 2 Sorts Of Honesty Requirements:

- Declarative Integrity limitations characterized utilizing CREATE& ALTER TABLE articulations.
- Procedural honesty requirements took care of by triggers.

A trigger is an instrument that is conjured when a specific activity happens on a specific table. Every trigger has 3 general parts:

- A name
- The activity
- The execution

SQL server keeps record of every change it makes to the db amid an exchange. This is essential in the event that a lapse happens amid the execution of the exchange. For this situation all already executed explanations inside of the exchange must be moved back. SQL server keeps every one of these records, specifically the previously, then after the fact values, in one or more documents called the exchange log. Each DB of the SQL server framework has its own particular exchange log. Concurrency in multi-client frameworks, for example, SQL Server has chosen impact of execution. At the point when access to the information is taken care of such that stand out project at once can utilize the information, preparing moderates significantly. SQL Server like all different DBMSs takes care of this issue utilizing exchanges. All announcements inside an exchange manufacture a nuclear unit. This implies that either all announcements are executed or for the situation of disappointment, all announcements are wiped out.

Elements of SQL Server

Microsoft SQL Server bolsters a full arrangement of elements that outcome in the accompanying. SQL incorporates an arrangement of managerial and advancement instruments that enhance our capacity to introduce, convey, oversee and use SQL Server over a few locales.

- **Adaptability**

The same database motor can be utilized crosswise over stages going from smart phones Microsoft Windows95 to substantial; multiprocessor servers running Microsoft Windows NT, Enterprise Edition.

- **Ease In Building Information Distribution Centers**

SQL Server incorporates instruments for removing and examining synopsis information for online investigative preparing (OLAP). SQL Server likewise incorporates apparatuses for outwardly planning databases and breaking down information utilizing English based inquiries.

SQL API (SQL Application Programming Interface)

Implanted SQL applications utilize the DB-library DLL to get to SQL server. The SQL Server ODBC driver clients don't get to Microsoft SQL Server straightforwardly. They utilize an application kept in touch with access the information in SQL Server. SQL Server can likewise be gotten to through COM, Microsoft ActiveX, or Windows DNA (Windows Distributed Internet Applications Architecture) parts. Applications are composed to get to SQL Server through a database Application Programming Interface (API).

Web Clients:

A Web customer comprises of two sections:

- Dynamic Web pages containing different sorts of markup dialect which are created by Web parts running in the Web level.
- Web program, which renders the pages got from the server.

A Web customer is now and again called a slim customer. Slim customers as a rule don't question databases, execute complex business guidelines, or associate with legacy applications.

HTML

HTML remains for Hyper Text Markup Language. It is a basic content designing dialect used to make hypertext records. It is a stage free dialect not at all like most other programming dialect. HTML is impartial and can be utilized on numerous stage or desktop.

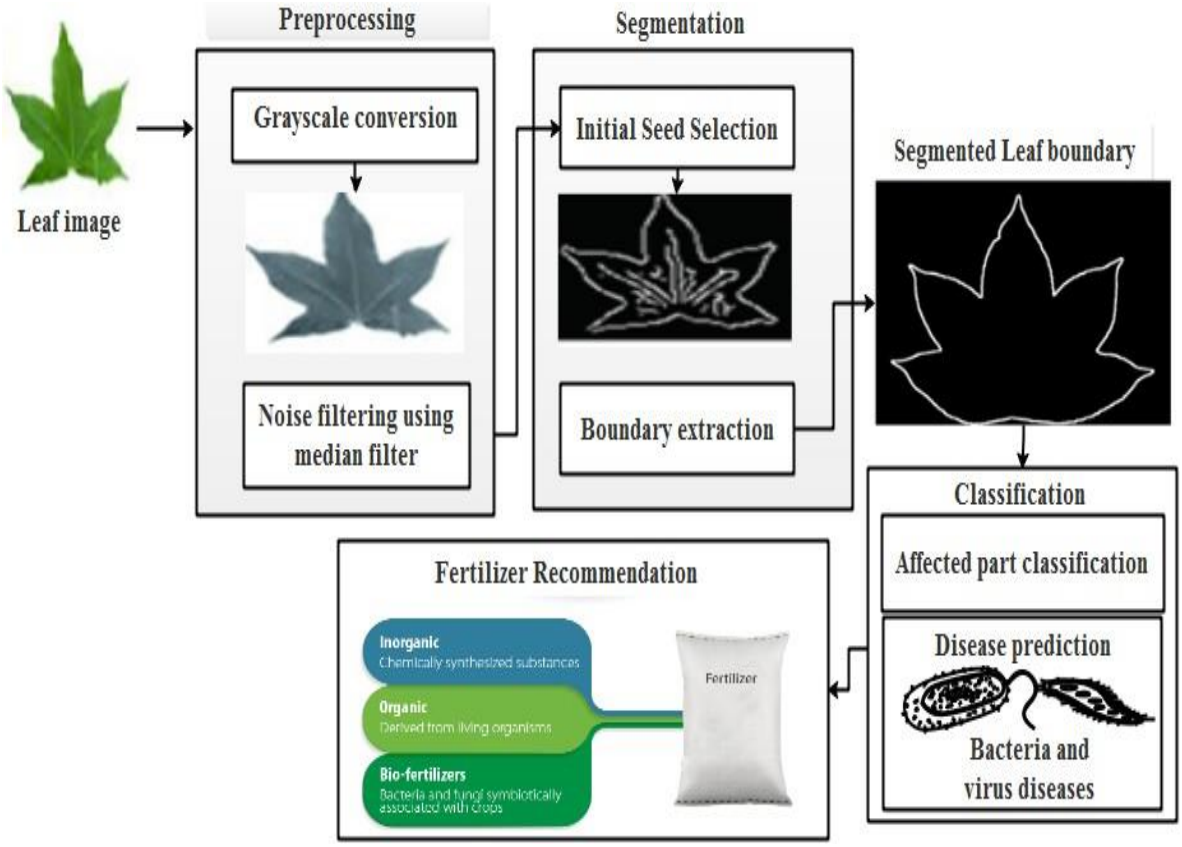
It is this component of HTML that makes it mainstream as standard on the WWW. This adaptable dialect permits the making of hypertext connections, otherwise called hyperlinks. These hyperlinks can be utilized to unite reports on diverse machine, on the same system or on an alternate system, or can even indicate purpose of content in the same record.

HTML is utilized for making archives where the accentuation is on the presence of the record. It is likewise utilized for DTP. The records made utilizing HTML can have content with diverse sizes, weights and hues. It can also contain graphics to make the document more effective.

SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE

System architecture involves the high level structure of software system abstraction, by using decomposition and composition, with architectural style and quality attributes. A software architecture design must conform to the major functionality and performance requirements of the system, as well as satisfy the non-functional requirements such as reliability, scalability, portability, and availability. System architecture must describe its group of components, their connections, interactions among them and deployment configuration of all components.

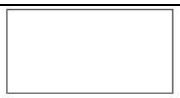
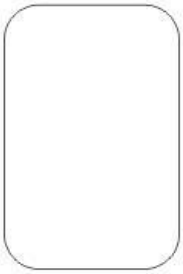


3.3 SYSTEM MODEL-DATA FLOW DIAGRAM

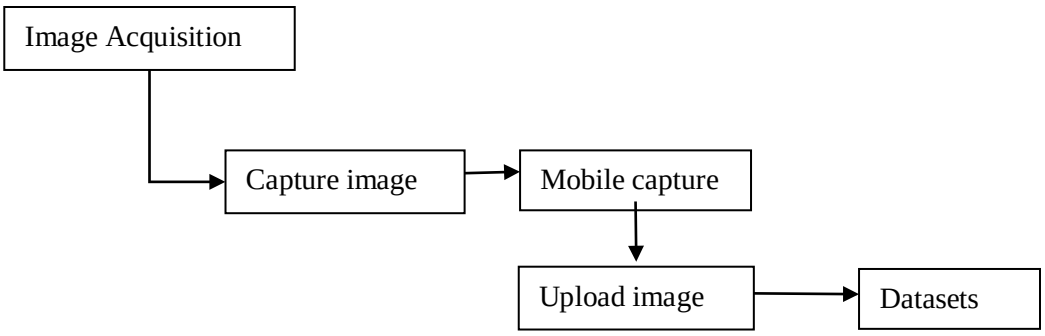
DATA FLOW DIAGRAM

A two-dimensional diagram explains how data is processed and transferred in a system. The graphical depiction identifies each source of data and how it interacts with other data sources to reach a common output. Individuals seeking to draft a data flow diagram must identify external inputs and outputs, determine how the inputs and outputs relate to each other, and explain with graphics how these connections relate and what they result in. This type of diagram helps business development and design teams visualize how data is processed and identify or improve certain aspects.

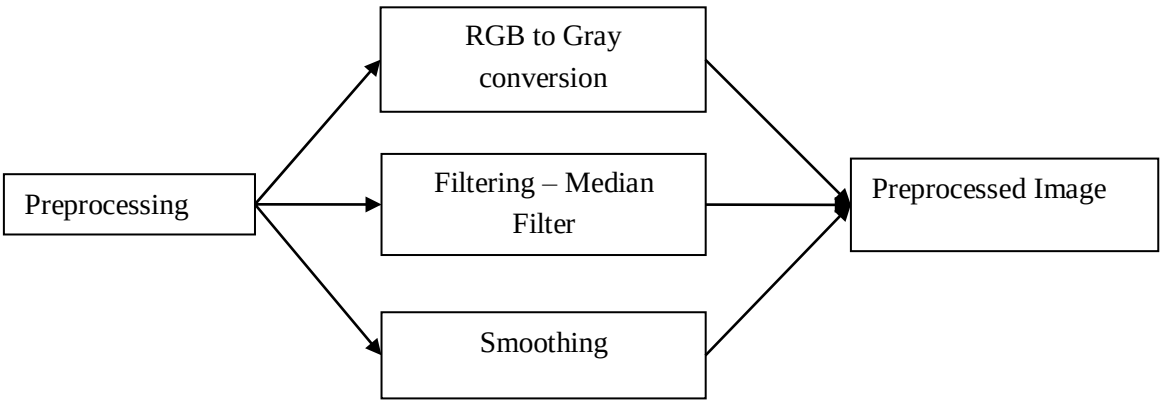
Data flow Symbols:

Symbol	**Description**
	An **entity**. A source of data or a destination for data.
	A **process** or task that is performed by the system.
	A **data store**, a place where data is held between processes.
	A **data flow**.

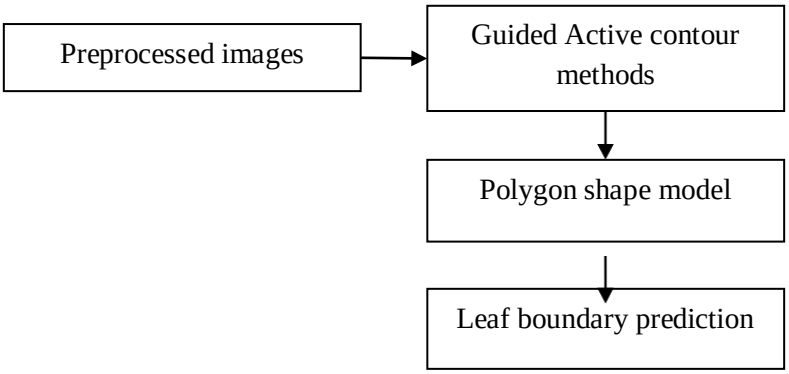
Level 0



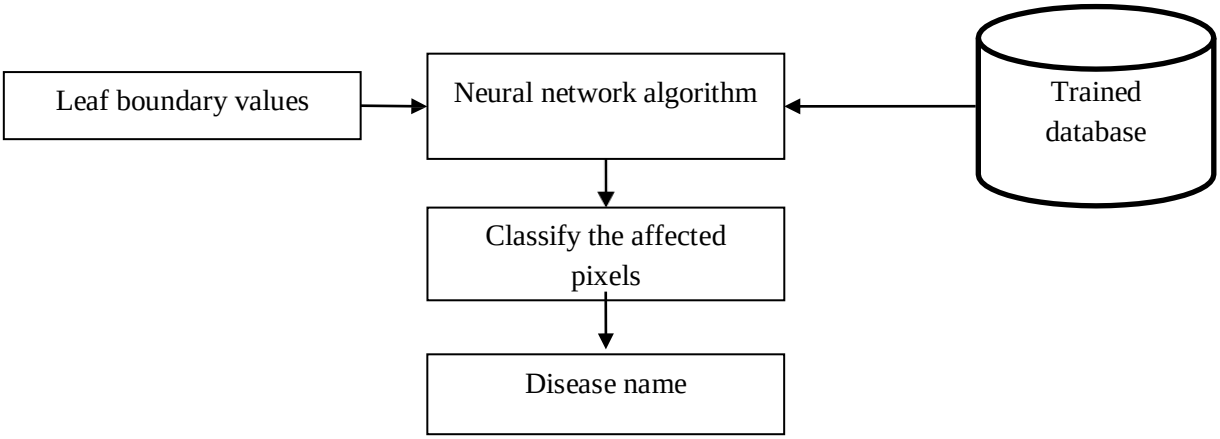
Level 1



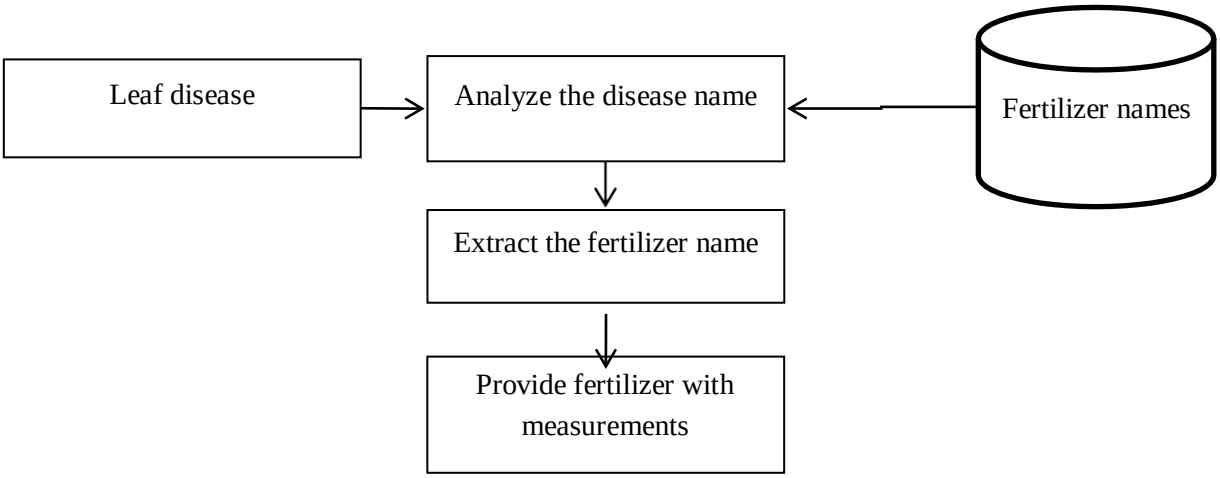
Level 2



Level 3



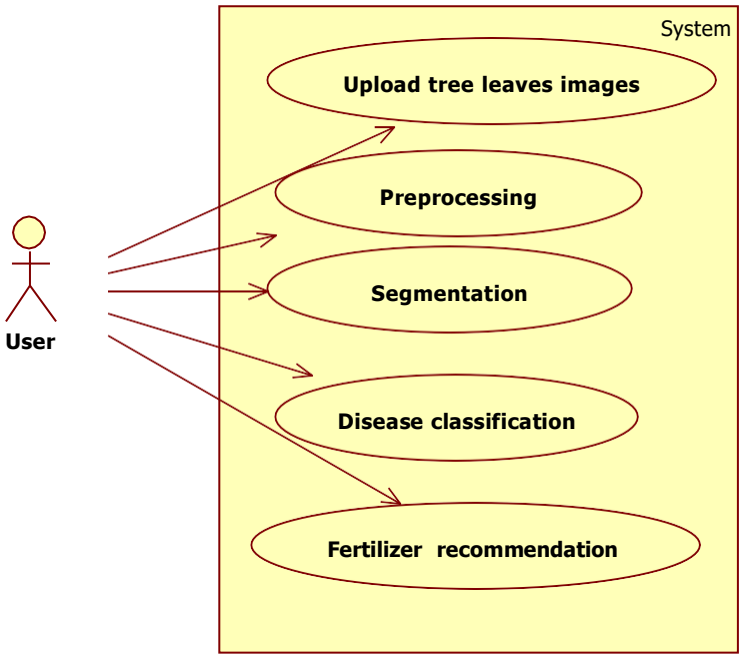
Level 4



UML DIAGRAM

4.1 UML DIAGRAM

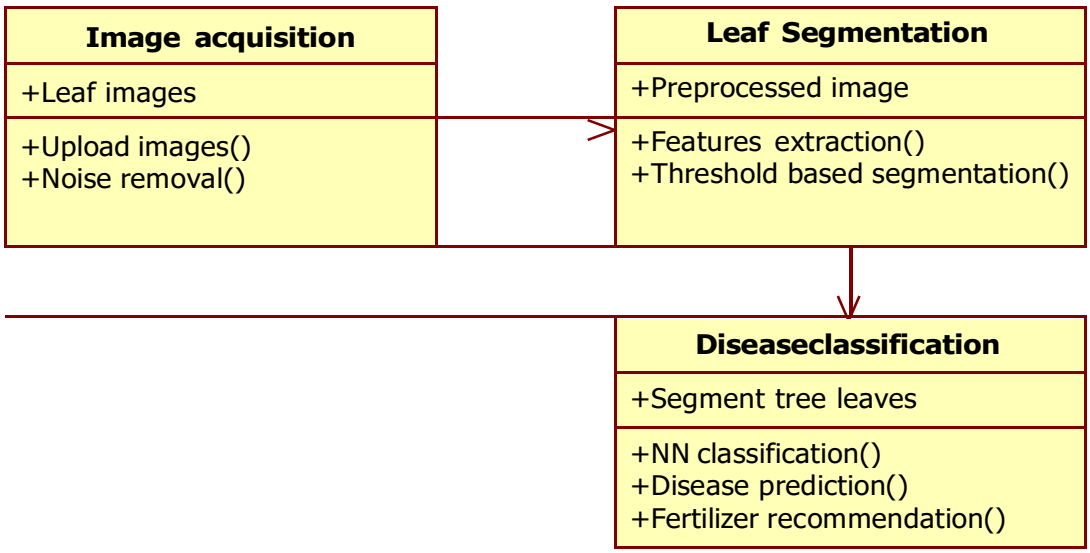
A use case diagram is a dynamic or behavior diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform.



4.1.2 CLASS DIAGRAM

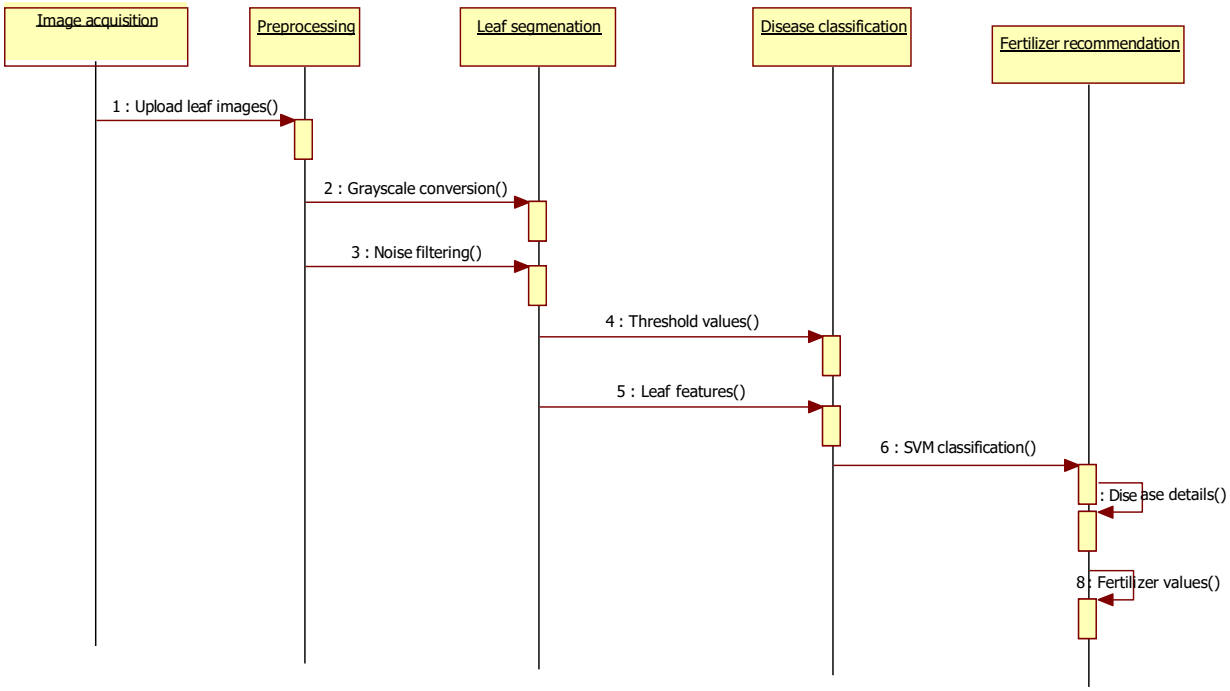
The class diagram is the main building block of object-oriented modeling. It is used both for general conceptual modeling of the systematics of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main elements,

interactions in the application, and the classes to be programmed.



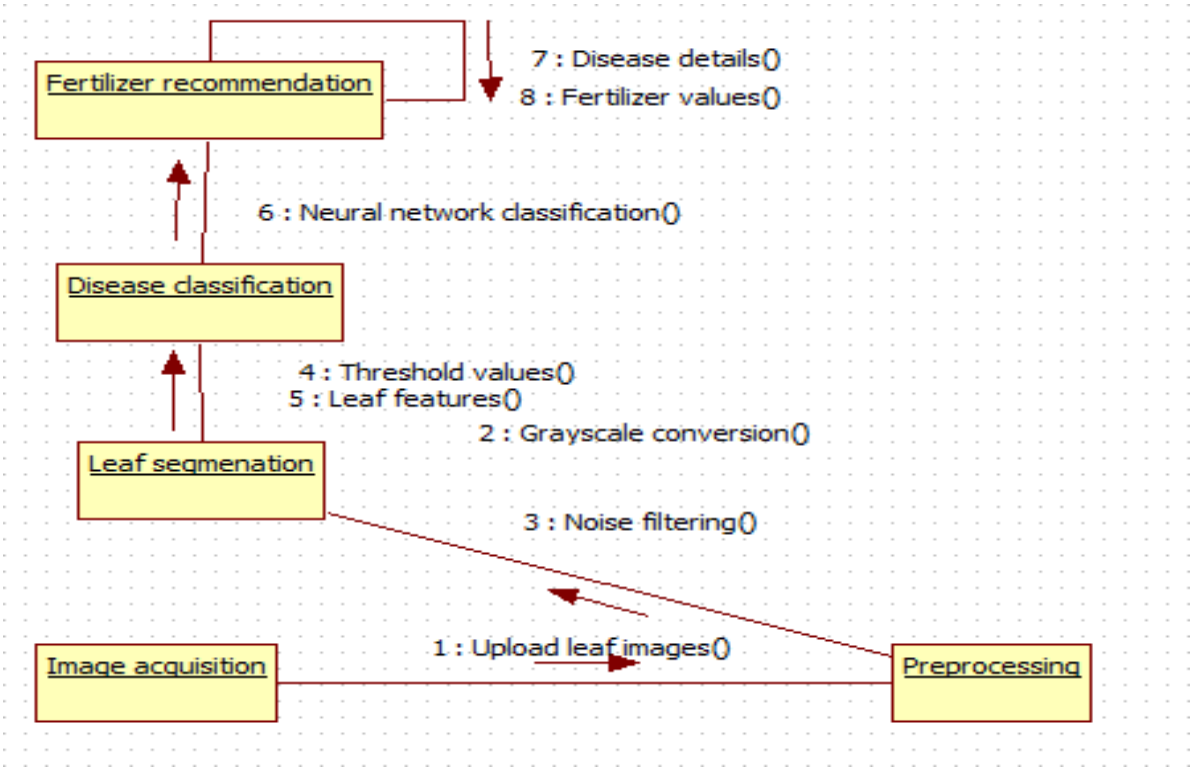
4.1.3 SEQUENCE DIAGRAM:

A Sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.



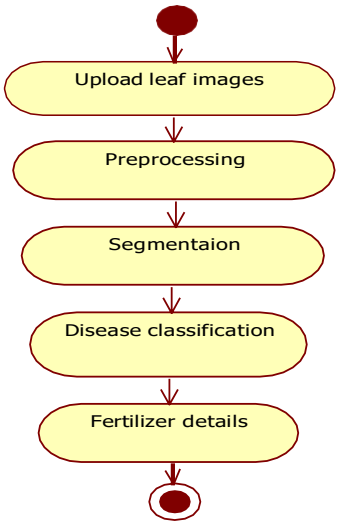
4.1.4 COLLABORATION DIAGRAM:

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object.Collaboration diagrams are created by first identifying the structural elements required to carry out the functionality of an interaction. A model is then built using the relationships between those elements. Several vendors offer software for creating and editing collaboration diagrams.



4.1.5 ACTIVITY DIAGRAM:

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. Activity diagram is basically a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagrams deals with all type of flow control by using different elements like fork, join etc.



5. IMPLEMENTATION

5.1 MODULE LIST

- Image Acquisition
- Preprocessing
- Image Segmentation
- Disease prediction
- Fertilizer Recommendation

5.2 MODULES DESCRIPTION

IMAGE ACQUISITION:

Leaves are structures specialized for photosynthesis and are arranged on the tree in such a way as to maximize their exposure to light without shading each other. In this module, we can upload the leaf images from the datasets. This database called LEAF was originally created for experiments with recognition of wood species based on a leaf shape. It contains leaves of species growing in the Czech Republic, both trees and bushes; native, invasive and imported (only those imported species which are common in parks are included). The number of samples (leaves) of one species varies from 2 to 25; their total number in the database is 795. The leaves were scanned with 300 dpi, threshold (binarized) , preprocessed (denoising and cleaning) and saved in PNG format.

PREPROCESSING:

In this module convert the RGB image into gray scale image. The colors of leaves are always green shades and the variety of changes in atmosphere cause the color feature having low reliability. Therefore, to recognize various plants using their leaves, the obtained leaf image in RGB format will be converted to gray scale before pre-processing. The formula used for converting the RGB pixel value to its grey scale counterpart is given in Equation.

$$Gray = 0.2989 * R + 0.5870 * G + 0.1140 * B$$

where R, G, B correspond to the color of the pixel, respectively.

Then remove the noises from images by using filter techniques. The goal of the filter is to filter out noise that has corrupted image. It is based on a statistical approach. Typical filters are designed for a desired frequency response. Filtering is a nonlinear operation often used in image processing to reduce "salt and pepper" noise.

IMAGE SEGMENTATION:

In this module, we can implement Guided active contour method with automatic descriptors. Unconstrained active contours applied to the complex natural images we aim at dealing with would produce unsatisfying contours, that would try and make their way through every possible gap and aw in the border of the leaf. The solution we propose is to use the polygonal model obtained after the first step not only as an initial leaf contour but also as a shape prior that will guide its evolution towards the real leaf boundary.

Use the resulting polygon as a shape prior to drive the evolution of an active contour

- Set the initial contour on a contracted version of the polygon
- Constraint the contour to remain close to the polygon

Energy Formulation

- For a contour τdelineating a region Ω(τ) :
- $E(\tau) = \alpha ELeaf(\tau) + \beta EShape(\tau) + \gamma EGradient(\tau) + \delta ESmooth(\tau) - \delta EBalloon(\tau)$

Instead of having an external energy term based on color consistency, or distance to a mean, we decided to reuse the dissimilarity map from the previous step, considering we have already an efficient measure of how well a pixel should fit in the leaf, in terms of color.

DISEASE PREDICTION:

Leaves are affected by bacteria, fungi, virus and other insects. In this module implement support vector machine algorithm to classify the leaf image as normal or affected. Vectors are constructed based leaf features such as color, shape, textures. Then hyperplane can be constructed with conditions to categorize the preprocessed leaves. And also implement multiclass classifier, we can predict diseases in leaf images with improved accuracy.

FERTILIZER RECOMMENDATION:

In this module recommend the fertilizer for affected leaves based on severity level. Fertilizers may be organic or inorganic. Admin can store the fertilizers based on disease categorization with severity levels. The measurements of fertilizers can be extracted based on disease severity.

6. CODING AND SYSTEM TESTING

6.1 CODING

The system moves onto the coding and testing phase after the design portion is completed. The real system is put into operation during the coding phase by translating the system's design into code written in a specific programming language. Therefore, whenever changes are needed, solid coding practises must be used so that the system can be quickly screwed.

6.2 SYSTEM TESTING:

Software testing is the last phase of the software development cycle. Testing is very important for the success of a system. System testing makes a logical assumption that if all parts of the system are correct, then the goal has been achieved. The testing should be done at the end of all development steps. Even though the final testing and verification are inevitable for better life and functionality of the software. The major phases in testing are design of test plan, setting up test case and test candidate and test procedure, testing and correction. This is a cycle process and the software will circulate through all the steps till it attains the required quality.

The testing is carried in the following steps,

1. Unit Testing

2. Validation Testing

3. System Testing

4. Acceptance Testing

5. Regression Testing

6. Database Testing

6.2.1 Unit Testing

Unit testing refers testing of all the individual programs. This is sometimes called as program testing. This test should be carried out during programming stage in order to find the errors in coding and logic for each program in each module. Unit test focuses verification

effort on the smallest unit of software design module. In this project, the user must fill each field otherwise the user to enter values.

6.2.2 Validation Testing

Valid and invalid data should be created and the program should be made to process this data to catch errors. When the user of each module wants to enter into the page by the login page using the use rid and password .If the user gives the wrong password or use rid then the information is provided to the user like “you must enter user id and password”. Here the inputs given by the user are validated. That is password validation, format of date are correct, textbox validation. Changes that need to be done after result of this testing.

6.2.3 Input Testing

Here system is tested with all variable combination of inputs. User may type data in situations like entering password, numerical details etc. The system is tested with all the cases and it responded with appropriate error messages.

6.2.4 Output Testing

Here the output is tested to view whether that the screen is what which is desired. It is also checked whether it is to the satisfaction of the user. Changes that need to be done can be done after the result is seen.

6.2.5 System Testing

System testing is used to test the entire system (Integration of all the modules). It also tests to find the discrepancies between the system and the original objective, current specification and system documentation. The entire system is checked to correct deviation to achieve correctness.

6.2.6 Acceptance Teasing

Acceptance testing is performed on a collection of business functions in a Production environment and after the completion of functional testing. This is the final Stage in the testing process before the system is accepted for operational use. This testing should be done with original data and with the presence of the users. This test confirms the system ready for production.

6.2.7 Regression Testing

Regression testing refers to the retesting components / functionality of the system to ensure that they function properly even after and change has been made to parts of the system. As detects are discovered in a component, modification are made to correct them.

6.2.8 Database Testing

The purpose of database testing is to determine how well the databases are meets requirements. This is an ongoing process because no database is static. When table is created, a mirror of the same should be created and stored. The original one should be left alone and its mirror images go through the various tests. This process continues until changes can be implemented in the original table.

6.3 FUNCTIONAL TESTING

Functional test can be defined as testing two or more modules together with the intent of finding defects, demonstrating that defects are not present, verifying that the module performs its intended functions as stated in the specification and establishing confidence that a program does what it is supposed to do.

6.3.1 WHITE BOX TESTING:

Testing based on an analysis of internal workings and structure of a piece of software. This testing can be done sing the percentage value of load and energy. The tester should know what exactly is done in the internal program. Includes techniques such as Branch Testing and Path Testing. Also known as Structural Testing and Glass Box Testing.

6.3.2 BLACK BOX TESTING:

Testing without knowledge of the internal workings of the item being tested. Tests are usually functional. This testing can be done by the user who has no knowledge of how the shortest path is found.

6.4 TESTING APPROACH

Software testing is a method of assessing the functionality of a software program. There are many different types of software testing but the two main categories are dynamic testing and static testing. Dynamic testing is an assessment that is conducted while the program is executed; static testing, on the other hand, is an examination of the program's code and associated documentation. Dynamic and static methods are often used together.

Testing is a set activity that can be planned and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it is vital success of the system.

- Testing Objectives:

There are several rules that can serve as testing objectives, they are

1. Testing is a process of executing a program with the intent of finding an error
2. A good test case is one that has high probability of finding an undiscovered error.
3. A successful test is one that uncovers an undiscovered error.

If testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrates that software functions appear to the working according to the specification, that performance requirements appear to have been met. Tests for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

Tests for implementation efficiency attempt to find ways to make a correct program faster or use less storage. It is a code-refining process, which reexamines the implementation phase of algorithm development. Tests for computational complexity amount to an experimental analysis of the complexity of an algorithm or an experimental comparison of two or more algorithms, which solve the same problem.

The data is entered in all forms separately and whenever an error occurred, it is corrected immediately. A quality team deputed by the management verified all the necessary documents and tested the Software while entering the data at all levels. The development process involves various types of testing. Each test type addresses a specific testing requirement. The most common types of testing involved in the development process are:

- Unit Test.
- System Test
- Integration Test
- Functional Test

6.5 TESTING

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the

configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

6.5.1 WHITE BOX TESTING

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

White Box Testing Techniques:

- **Statement Coverage -** This technique is aimed at exercising all programming statements with minimal tests.
- **Branch Coverage -** This technique is running a series of tests to ensure that all branches are tested at least once.
- **Path Coverage -** This technique corresponds to testing all possible paths which means that each statement and branch is covered.

6.5.2 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.

Black box testing focuses on the functional requirements of the software. That is black box testing enables the software engineer to drive a set of input conditions that will fully exercise the requirements for a program. Black box testing is not an alternative for white box testing techniques. Rather, it is a complementary approach that is likely to uncover different class of errors. Black box testing attempts to find errors in the following categories:

- Interface errors.
- Performances in data structures or external database access.

- Performance errors.
- Initialization and termination errors.
- Incorrect or missing functions.

6.5.3 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.

- Reduces Defects in the newly developed features or reduces bugs when changing the existing functionality.
- Reduces Cost of Testing as defects are captured in very early phase.
- Improves design and allows better refactoring of code.
- Unit Tests, when integrated with build gives the quality of the build as well.

6.5.4 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.

6.5.5 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.

6.5.6 ACCEPTANCE TESTING

Acceptance testing can be defined in many ways, but a simple definition is the succeeds when the software functions in a manner that can be reasonable expected by the customer. After the acceptance test has been conducted, one of the two possible conditions exists. This is to fine whether the inputs are accepted by the database or other validations. For example accept only numbers in the numeric field, date format data in the date field. Also the null check for the not null fields. If any error occurs then show the error messages. The function of performance characteristics to specification and is accepted. A deviation from specification is uncovered and a deficiency list is created. 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.

7.1 CONCLUSION

We presented a machine learning approach for crop yield prediction, which demonstrated superior performance in Crop Challenge using large datasets of products. The approach used deep neural networks to make yield predictions (including yield, check yield, and yield difference) based on genotype and environment data. The carefully designed deep neural networks were able to learn nonlinear and complex relationships between genes, environmental conditions, as well as their interactions from historical data and make reasonably accurate predictions of yields for new hybrids planted in new locations with known weather conditions. Performance of the model was found to be relatively sensitive to the quality of weather prediction, which suggested the importance of weather prediction techniques. We trained two deep neural networks, one for yield and the other for check yield, and then used the difference of their outputs as the prediction for yield difference. This model structure was found to be more effective than using one single neural network for yield difference, because the genotype and environment effects are more directly related to the yield and check yield than their difference. In modern era, the deep neural network is the prominent tool in agricultural industry for providing support to farmers in monitoring crop yield based on multiple parameters. Thus, the machine learning model provides high accuracy in detecting the suitable crop identification compared to other methodologies.

7.2 FUTURE WORK

This project describes crop yield prediction ability of the algorithm. In future we can determine the efficient algorithm based on their accuracy metrics that will helps to choose an efficient algorithm for crop yield prediction

APPENDIX 1

SAMPLE CODE

```
import tensorflow as tf
import  time
import numpy as np
import os

start = time.time()
#try:
# Total iterations
final_iter = 1000

# Assign the batch value
batch_size = 20

# 20% of the data will automatically be used for validation
validation_size = 0.2
img_size = 128
num_channels = 3
train_path = r'data\Train'

# Prepare input data
if not os.path.exists(train_path):
print("No such directory")
raise Exception
classes = os.listdir(train_path)
num_classes = len(classes)

# We shall load all the training and validation images and labels into memory
using openCV and use that during training
data = dataset.read_train_sets(train_path, img_size, classes,
validation_size=validation_size)

# Display the stats
print("Complete reading input data. Will Now print a snippet of it")
print("Number of files in Training-set:\t\t{}".format(len(data.train.labels)))
print("Number of files in Validation-set:\t{}".format(len(data.valid.labels)))
session = tf.compat.v1.Session()
x = tf.compat.v1.placeholder(tf.float32, shape=[None, img_size, img_size,
num_channels], name='x')

## labels
y_true = tf.compat.v1.placeholder(tf.float32, shape=[None, num_classes],
name='y_true')
y_true_cls = tf.argmax(y_true, dimension=1)

##Network graph params
filter_size_conv1 = 3
num_filters_conv1 = 32

filter_size_conv2 = 3
num_filters_conv2 = 32

filter_size_conv3 = 3
num_filters_conv3 = 64

fc_layer_size = 128
```

```
def create_weights(shape):
return tf.Variable(tf.random.truncated_normal(shape, stddev=0.05))


def create_biases(size):
return tf.Variable(tf.constant(0.05, shape=[size]))



def make_generator_model(input,
                                num_input_channels,
                                conv_filter_size,
                                num_filters):
## We shall define the weights that will be trained using create_weights function.
weights = create_weights(shape=[conv_filter_size, conv_filter_size,
num_input_channels, num_filters])
## We create biases using the create_biases function. These are also trained.
biases = create_biases(num_filters)

## Creating the convolutional layer
layer = tf.nn.conv2d(input=input,
filter=weights,
strides=[1, 1, 1, 1],
padding='SAME')

    layer += biases

## We shall be using max-pooling.
layer = tf.nn.max_pool(value=layer,
ksize=[1, 2, 2, 1],
strides=[1, 2, 2, 1],
padding='SAME')
## Output of pooling is fed to Relu which is the activation function for us.
layer = tf.nn.relu(layer)

return layer


# Function to create a Flatten Layer
def create_flatten_layer(layer):
# We know that the shape of the layer will be [batch_size img_size img_size
num_channels]
    # But let's get it from the previous layer.
layer_shape = layer.get_shape()

## Number of features will be img_height * img_width* num_channels. But we shall
calculate it in place of hard-coding it.
num_features = layer_shape[1:4].num_elements()

## Now, we Flatten the layer so we shall have to reshape to num_features
layer = tf.reshape(layer, [-1, num_features])

return layer


# Function to create a Fully - Connected Layer
def create_fc_layer(input,
```

```
                    num_inputs,
                    num_outputs,
                    use_relu=True):
# Let's define trainable weights and biases.
weights = create_weights(shape=[num_inputs, num_outputs])
    biases = create_biases(num_outputs)

# Fully connected layer takes input x and produces wx+b.Since, these are matrices,
we use matmul function in Tensorflow
layer = tf.matmul(input, weights) + biases
if use_relu:
        layer = tf.nn.relu(layer)

return layer


# Create all the layers
layer_conv1 = make_generator_model(input=x,
num_input_channels=num_channels,
conv_filter_size=filter_size_conv1,
num_filters=num_filters_conv1)
layer_conv2 = make_generator_model(input=layer_conv1,
num_input_channels=num_filters_conv1,
conv_filter_size=filter_size_conv2,
num_filters=num_filters_conv2)

layer_conv3 = make_generator_model(input=layer_conv2,
num_input_channels=num_filters_conv2,
conv_filter_size=filter_size_conv3,
num_filters=num_filters_conv3)

layer_flat = create_flatten_layer(layer_conv3)

layer_fc1 = create_fc_layer(input=layer_flat,
num_inputs=layer_flat.get_shape()[1:4].num_elements(),
num_outputs=fc_layer_size,
use_relu=True)

layer_fc2 = create_fc_layer(input=layer_fc1,
num_inputs=fc_layer_size,
num_outputs=num_classes,
use_relu=False)

y_pred = tf.nn.softmax(layer_fc2, name='y_pred')

y_pred_cls = tf.argmax(y_pred, dimension=1)
session.run(tf.compat.v1.global_variables_initializer())
cross_entropy = tf.nn.softmax_cross_entropy_with_logits_v2(logits=layer_fc2,
labels=y_true)
cost = tf.reduce_mean(cross_entropy)
optimizer = tf.compat.v1.train.AdamOptimizer(learning_rate=1e-4).minimize(cost)
correct_prediction = tf.equal(y_pred_cls, y_true_cls)
accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))

session.run(tf.compat.v1.global_variables_initializer())


# Display all stats for every epoch
def show_progress(epoch, feed_dict_train, feed_dict_validate, val_loss,
```

```
total_epochs):
    acc = session.run(accuracy, feed_dict=feed_dict_train)
    val_acc = session.run(accuracy, feed_dict=feed_dict_validate)
    msg = "Training Epoch {0}/{4} --- Training Accuracy: {1:>6.1%}, Validation
Accuracy: {2:>6.1%},  Validation Loss: {3:.3f}"
print(msg.format(epoch + 1, acc, val_acc, val_loss, total_epochs))


total_iterations = 0

saver = tf.compat.v1.train.Saver()

print("")


# Training Function
def train(num_iteration):
global total_iterations

for i in range(total_iterations,
                   total_iterations + num_iteration):

        x_batch, y_true_batch, _, cls_batch = data.train.next_batch(batch_size)
        x_valid_batch, y_valid_batch, _, valid_cls_batch =
data.valid.next_batch(batch_size)

        feed_dict_tr = {x: x_batch,
                        y_true: y_true_batch}
        feed_dict_val = {x: x_valid_batch,
                         y_true: y_valid_batch}

        session.run(optimizer, feed_dict=feed_dict_tr)

if i % int(data.train.num_examples / batch_size) == 0:
            val_loss = session.run(cost, feed_dict=feed_dict_val)
            epoch = int(i / int(data.train.num_examples / batch_size))
# print(data.train.num_examples)
            # print(batch_size)
            # print(int(data.train.num_examples/batch_size))
            # print(i)

total_epochs = int(num_iteration / int(data.train.num_examples / batch_size)) + 1
show_progress(epoch, feed_dict_tr, feed_dict_val, val_loss, total_epochs)
            saver.save(session, 'trained_model')

    total_iterations += num_iteration


train(num_iteration=final_iter)


#except Exception as e:
    #print("Exception:",e)

# Calculate execution time
end = time.time()
dur = end-start
print("")
if dur<60:
```

```
print("Execution Time:",dur,"seconds")
elif dur>60 and dur<3600:
    dur=dur/60
print("Execution Time:",dur,"minutes")
else:
    dur=dur/(60*60)
print("Execution Time:",dur,"hours")
from flask import Flask, render_template, flash, request, session,send_file
from flask import render_template, redirect, url_for, request
import warnings
import datetime
import cv2
import tensorflow as tf
import numpy as np

from tkinter import *
import os


app = Flask(_name_)
app.config['DEBUG']
app.config['SECRET_KEY'] = '7d441f27d441f27567d441f2b6176a'

@app.route("/")
def homepage():

return render_template('index.html')




@app.route("/Test")
def Test():
return render_template('Test.html')



@app.route("/train", methods=['GET', 'POST'])
def train():
if request.method == 'POST':
import model as model

return render_template('Tranning.html')



@app.route("/testimage", methods=['GET', 'POST'])
def testimage():
if request.method == 'POST':


        file = request.files['fileupload']
        file.save('data/alien_test/Test.jpg')
```

```
img = cv2.imread('data/alien_test/Test.jpg')



        train_path = r'data\train'
if not os.path.exists(train_path):
print("No such directory")
raise Exception
# Path of testing images
dir_path  =  r'data\alien_test'
if not os.path.exists(dir_path):
print("No  such  directory")
raise Exception

# Walk though all testing images one by one
for root, dirs, files in os.walk(dir_path):
for name in files:

print("")
                image_path = name
                filename = dir_path + '\\' + image_path
print(filename)
                image_size = 128
num_channels = 3
images = []

if os.path.exists(filename):

# Reading the image using OpenCV
image1 = cv2.imread(filename)

                    import_file_path = filename

                    image = cv2.imread(import_file_path)
                    fnm = os.path.basename(import_file_path)
                    filename = 'Test.jpg'
cv2.imwrite(filename, image)
# print("After saving image:")

print("\n**********************\nImage : " + fnm + "\n**********************")
                    img = cv2.imread(import_file_path)
if img is None:
print('no data')

                    img1 = cv2.imread(import_file_path)
print(img.shape)
                    img = cv2.resize(img, ((int)(img.shape[1] / 5),
(int)(img.shape[0] / 5)))
                    original = img.copy()
                    neworiginal = img.copy()
                    cv2.imshow('original', img1)
                    gray = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)

                    cv2.imshow('Original image', img1)
                    orimage = 'static/Out/Test.jpg'
cv2.imwrite(orimage, img1)


                    cv2.imshow('Gray image', gray)
```

```
                    gry = 'static/Out/gry.jpg'

cv2.imwrite(gry, gray)


                    p = 0
for i in range(img.shape[0]):

for j in range(img.shape[1]):
                            B = img[i][j][0]
                            G = img[i][j][1]
                            R = img[i][j][2]
if (B >110 and G >110 and R >110):
                                p += 1

totalpixels = img.shape[0] * img.shape[1]
                    per_white = 100 * p / totalpixels
if per_white >10:
                        img[i][j] = [500, 300, 200]
                        cv2.imshow('color change', img)
# Guassian blur
blur1 = cv2.GaussianBlur(img, (3, 3), 1)
# mean-shift algo
newimg = np.zeros((img.shape[0], img.shape[1], 3), np.uint8)
                    criteria = (cv2.TERM_CRITERIA_EPS +
cv2.TERM_CRITERIA_MAX_ITER, 10, 1.0)
                    img = cv2.pyrMeanShiftFiltering(blur1, 20, 30, newimg, 0,
criteria)
                    cv2.imshow('means shift image', img)

                    noise = 'static/Out/noise.jpg'

cv2.imwrite(noise, img)


# Guassian blur
blur = cv2.GaussianBlur(img, (11, 11), 1)

                    blur = cv2.GaussianBlur(img, (11, 11), 1)
# Canny-edge detection
canny = cv2.Canny(blur, 160, 290)
                    canny = cv2.cvtColor(canny, cv2.COLOR_GRAY2BGR)
# contour to find leafs
bordered = cv2.cvtColor(canny, cv2.COLOR_BGR2GRAY)
                    contours, hierarchy = cv2.findContours(bordered,
cv2.RETR_TREE, cv2.CHAIN_APPROX_NONE)
                    maxC = 0
for x in range(len(contours)):
if len(contours[x]) > maxC:
                            maxC = len(contours[x])
                            maxid = x
perimeter = cv2.arcLength(contours[maxid], True)
# print perimeter
Tarea = cv2.contourArea(contours[maxid])
                    cv2.drawContours(neworiginal, contours[maxid], -1, (0, 0,
255))
                    cv2.imshow('Contour', neworiginal)
# cv2.imwrite('Contour complete leaf.jpg',neworiginal)
```

```
                    # Creating rectangular roi around contour
height, width, _ = canny.shape
                    min_x, min_y = width, height
                    max_x = max_y = 0
frame = canny.copy()
# computes the bounding box for the contour, and draws it on the frame,
for contour, hier in zip(contours, hierarchy):
                        (x, y, w, h) = cv2.boundingRect(contours[maxid])
                        min_x, max_x = min(x, min_x), max(x + w, max_x)
                        min_y, max_y = min(y, min_y), max(y + h, max_y)
if w >80 and h >80:
# cv2.rectangle(frame, (x,y), (x+w,y+h), (255, 0, 0), 2)   #we do not draw the
rectangle as it interferes with contour later on
roi = img[y:y + h, x:x + w]
                            originalroi = original[y:y + h, x:x + w]
if (max_x - min_x >0 and max_y - min_y >0):
                        roi = img[min_y:max_y, min_x:max_x]
                        originalroi = original[min_y:max_y, min_x:max_x]
                        cv2.rectangle(frame, (min_x, min_y), (max_x, max_y), (255,
0, 0),
2)  # we do not draw the rectangle as it interferes with contour
cv2.imshow('ROI', frame)

                    roi12 = 'static/Out/roi.jpg'

cv2.imwrite(roi12, frame)



                    cv2.imshow('rectangle ROI', roi)
img = roi
# Changing colour-space
                    # imghsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
 imghls = cv2.cvtColor(roi, cv2.COLOR_BGR2HLS)
                    cv2.imshow('HLS', imghls)
                    imghls[np.where((imghls == [30, 200, 2]).all(axis=2))] = [0,
200, 0]
                    cv2.imshow('new HLS', imghls)
# Only hue channel
huehls = imghls[:, :, 0]
                    cv2.imshow('img_hue hls', huehls)
# ret, huehls = cv2.threshold(huehls,2,255,cv2.THRESH_BINARY)
huehls[np.where(huehls == [0])] = [35]
                    cv2.imshow('img_hue with my mask', huehls)
# Thresholding on hue image
ret, thresh = cv2.threshold(huehls, 28, 255, cv2.THRESH_BINARY_INV)
                    cv2.imshow('thresh', thresh)
# Masking thresholded image from original image
mask = cv2.bitwise_and(originalroi, originalroi, mask=thresh)
                    cv2.imshow('masked out img', mask)

# Resizing the image to our desired size and preprocessing will be done exactly as
done during training
image = cv2.resize(image1, (image_size, image_size), 0, 0, cv2.INTER_LINEAR)
                    images.append(image)
                    images = np.array(images, dtype=np.uint8)
                    images = images.astype('float32')
                    images = np.multiply(images, 1.0 / 255.0)
```

```
# The input to the network is of shape [None image_size image_size num_channels].
Hence we reshape.
x_batch = images.reshape(1, image_size, image_size, num_channels)

# Let us restore the saved model
sess = tf.compat.v1.Session()
# Step-1: Recreate the network graph. At this step only graph is created.
saver = tf.compat.v1.train.import_meta_graph('models/trained_model.meta')
# Step-2: Now let's load the weights saved using the restore method.
saver.restore(sess, tf.train.latest_checkpoint('./models/'))

# Accessing the default graph which we have restored
graph = tf.compat.v1.get_default_graph()

# Now, let's get hold of the op that we can be processed to get the output.
                    # In the original network y_pred is the tensor that is the
prediction of the network
y_pred = graph.get_tensor_by_name("y_pred:0")

## Let's feed the images to the input placeholders
x = graph.get_tensor_by_name("x:0")
                    y_true = graph.get_tensor_by_name("y_true:0")
                    y_test_images = np.zeros((1, len(os.listdir(train_path))))

# Creating the feed_dict that is required to be fed to calculate y_pred
feed_dict_testing = {x: x_batch, y_true: y_test_images}
                    result = sess.run(y_pred, feed_dict=feed_dict_testing)
# Result is of this format [[probabiliy_of_classA probability_of_classB .... ]]
print(result)

# Convert np.array to list
a = result[0].tolist()
                    r = 0

# Finding the maximum of all outputs
max1 = max(a)
                    index1 = a.index(max1)
                    predicted_class = None

# Walk through directory to find the label of the predicted output
count = 0
for root, dirs, files in os.walk(train_path):
for name in dirs:
if count == index1:
                                predicted_class = name
                            count += 1

# If the maximum confidence output is largest of all by a big margin then
                    # print the class or else print a warning
for i in a:
if i != max1:
if max1 - i < i:
                                r = 1

out = ''

pre = ""
if r == 0:
print(predicted_class)
```

```
if (predicted_class == "Black spot"):
                            out = predicted_class

                            pre = 'Griffin  Fertilizer  reducing the fungus'

elif (predicted_class == "canker"):
                            out = predicted_class
                            pre = 'sprayed with Bordeaux mixture 1.0 per cent.'


elif (predicted_class == "greening"):
                            out =  predicted_class
                            pre =  'Mn-Zn-Fe-B micronutrient fertilizer'

elif (predicted_class == "healthy"):
                            out =  predicted_class
# messagebox.showinfo("Uses", '')
elif (predicted_class == "Melanose"):
                            out = predicted_class
                            pre =  'strobilurin fungicide'




else:

                        out = 'Could not classify with definite confidence'


else:
print("File does not exist")







        org = 'static/Out/Test.jpg'
gry ='static/Out/gry.jpg'
noise = 'static/Out/noise.jpg'
roi12 = 'static/Out/roi.jpg'




return
render_template('Test.html',result=out,org=org,gry=gry,inv=noise,noi=roi12,fer=pre
)



def sendmsg(targetno,message):
import requests

requests.post("http://smsserver9.creativepoint.in/api.php?username=fantasy&passwor
d=596692&to=" + targetno + "&from=FSSMSS&message=Dear user  your msg is " +
```

```
message + " Sent By FSMSG
FSSMSS&PEID=1501563800000030506&templateid=1507162882948811640")
```

```
if__name__== '__main__':
    app.run(debug=True, use_reloader=True)
import cv2
import os
import glob
from sklearn.utils import shuffle
import numpy as np


def load_train(train_path, image_size, classes):
    images = []
    labels = []
    img_names = []
    cls = []

print('Going to read training images')
for fields in classes:
        index = classes.index(fields)
print('Now going to read {} files (Index: {})'.format(fields, index))
        path = os.path.join(train_path, fields, '*g')
        files = glob.glob(path)
for fl in files:
            image = cv2.imread(fl)
            image = cv2.resize(image, (image_size, image_size),0,0,
cv2.INTER_LINEAR)
            image = image.astype(np.float32)
            image = np.multiply(image, 1.0 / 255.0)
            images.append(image)
            label = np.zeros(len(classes))
            label[index] = 1.0
labels.append(label)
            flbase = os.path.basename(fl)
            img_names.append(flbase)
            cls.append(fields)
    images = np.array(images)
    labels = np.array(labels)
    img_names = np.array(img_names)
    cls = np.array(cls)

return images, labels, img_names, cls


class DataSet(object):

def_init_(self, images, labels, img_names, cls):
self._num_examples = images.shape[0]
```

```
self._images = images
self._labels = labels
self._img_names = img_names
self._cls = cls
self._epochs_done = 0
self._index_in_epoch = 0

@property
def images(self):
return self._images

@property
def labels(self):
return self._labels

@property
def img_names(self):
return self._img_names

@property
def cls(self):
return self._cls

@property
def num_examples(self):
return self._num_examples

@property
def epochs_done(self):
return self._epochs_done

def next_batch(self, batch_size):
"""Return the next `batch_size` examples from this data set."""
start = self._index_in_epoch
self._index_in_epoch += batch_size

if self._index_in_epoch >self._num_examples:
# After each epoch we update this
self._epochs_done += 1
start = 0
self._index_in_epoch = batch_size
assert batch_size <= self._num_examples
    end = self._index_in_epoch

return self._images[start:end], self._labels[start:end],
self._img_names[start:end], self._cls[start:end]


def read_train_sets(train_path, image_size, classes, validation_size):
class DataSets(object):
pass
data_sets = DataSets()

  images, labels, img_names, cls = load_train(train_path, image_size, classes)
  images, labels, img_names, cls = shuffle(images, labels, img_names, cls)

if isinstance(validation_size, float):
    validation_size = int(validation_size * images.shape[0])
```

```
  validation_images = images[:validation_size]
  validation_labels = labels[:validation_size]
  validation_img_names = img_names[:validation_size]
  validation_cls = cls[:validation_size]

  train_images = images[validation_size:]
  train_labels = labels[validation_size:]
  train_img_names = img_names[validation_size:]
  train_cls = cls[validation_size:]

  data_sets.train = DataSet(train_images, train_labels, train_img_names,
train_cls)
  data_sets.valid = DataSet(validation_images, validation_labels,
validation_img_names, validation_cls)

return  data_sets
import tensorflow as tf
import numpy as np

from tkinter import *
import os
from tkinter import filedialog
import cv2
import time
from matplotlib import pyplot as plt
from tkinter import messagebox




def endprogram():
print ("\nProgram terminated!")
   sys.exit()







def training():

import Training as tr




def imgtraining():
    import_file_path = filedialog.askopenfilename()

    image = cv2.imread(import_file_path)
    filename = 'Test.jpg'
```

```
cv2.imwrite(filename, image)
print("After saving image:")

    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    cv2.imshow('Original image', image)
    cv2.imshow('Gray image', gray)
# import_file_path = filedialog.askopenfilename()
print(import_file_path)
    fnm = os.path.basename(import_file_path)
print(os.path.basename(import_file_path))

from PIL import Image, ImageOps

    im = Image.open(import_file_path)
    im_invert = ImageOps.invert(im)
    im_invert.save('lena_invert.jpg', quality=95)
    im = Image.open(import_file_path).convert('RGB')
    im_invert = ImageOps.invert(im)
    im_invert.save('tt.png')
    image2 = cv2.imread('tt.png')
    cv2.imshow("Invert", image2)

"""----------------------------------------------"""

img = image

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    cv2.imshow('Original image', img)
#cv2.imshow('Gray image', gray)
dst = cv2.fastNlMeansDenoisingColored(img, None, 10, 10, 7, 21)
    cv2.imshow("Nosie Removal", dst)

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)



print("\n*********************\nImage : " + fnm + "\n*********************")
    img = cv2.imread(import_file_path)
if img is None:
print('no data')

    img1 = cv2.imread(import_file_path)
print(img.shape)
    img = cv2.resize(img, ((int)(img.shape[1] / 5), (int)(img.shape[0] / 5)))
original = img.copy()
neworiginal = img.copy()
    cv2.imshow('original', img1)
    gray = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)

    cv2.imshow('Original image', img1)
# cv2.imshow('Gray image', gray)
p = 0
for i in range(img.shape[0]):

for j in range(img.shape[1]):
            B = img[i][j][0]
            G = img[i][j][1]
            R = img[i][j][2]
```

```
if (B >110 and G >110 and R >110):
                p += 1

totalpixels = img.shape[0] * img.shape[1]
    per_white = 100 * p / totalpixels
if per_white >10:
        img[i][j] = [500, 300, 200]
        cv2.imshow('color change', img)
# Guassian blur
blur1 = cv2.GaussianBlur(img, (3, 3), 1)
# mean-shift algo
newimg = np.zeros((img.shape[0], img.shape[1], 3), np.uint8)
    criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 10, 1.0)
    img = cv2.pyrMeanShiftFiltering(blur1, 20, 30, newimg, 0, criteria)
    cv2.imshow('means shift image', img)
# Guassian blur
blur = cv2.GaussianBlur(img, (11, 11), 1)
    cv2.imshow('Noise Remove', blur)
    corners = cv2.goodFeaturesToTrack(gray, 27, 0.01, 10)
    corners = np.int0(corners)

# we iterate through each corner,
    # making a circle at each point that we think is a corner.
for i in corners:
        x, y = i.ravel()
        cv2.circle(image, (x, y), 3, 255, -1)

    plt.imshow(image), plt.show()
```

```
def testing():
global testing_screen
    testing_screen = Toplevel(main_screen)
    testing_screen.title("Testing")
# login_screen.geometry("400x300")
testing_screen.geometry("600x450+650+150")
    testing_screen.minsize(120, 1)
    testing_screen.maxsize(1604, 881)
    testing_screen.resizable(1, 1)
# login_screen.title("New Toplevel")

Label(testing_screen, text='''Upload Image''', background="#d9d9d9",
disabledforeground="#a3a3a3",
foreground="#000000", bg="turquoise", width="300", height="2", font=("Calibri",
16)).pack()
    Label(testing_screen, text="").pack()
    Label(testing_screen, text="").pack()
    Label(testing_screen, text="").pack()
    Button(testing_screen, text='''Upload Image''', font=(
'Verdana', 15), height="2", width="30", command=imgtest).pack()

def imgtest():
    import_file_path = filedialog.askopenfilename()

    image = cv2.imread(import_file_path)
print(import_file_path)
```

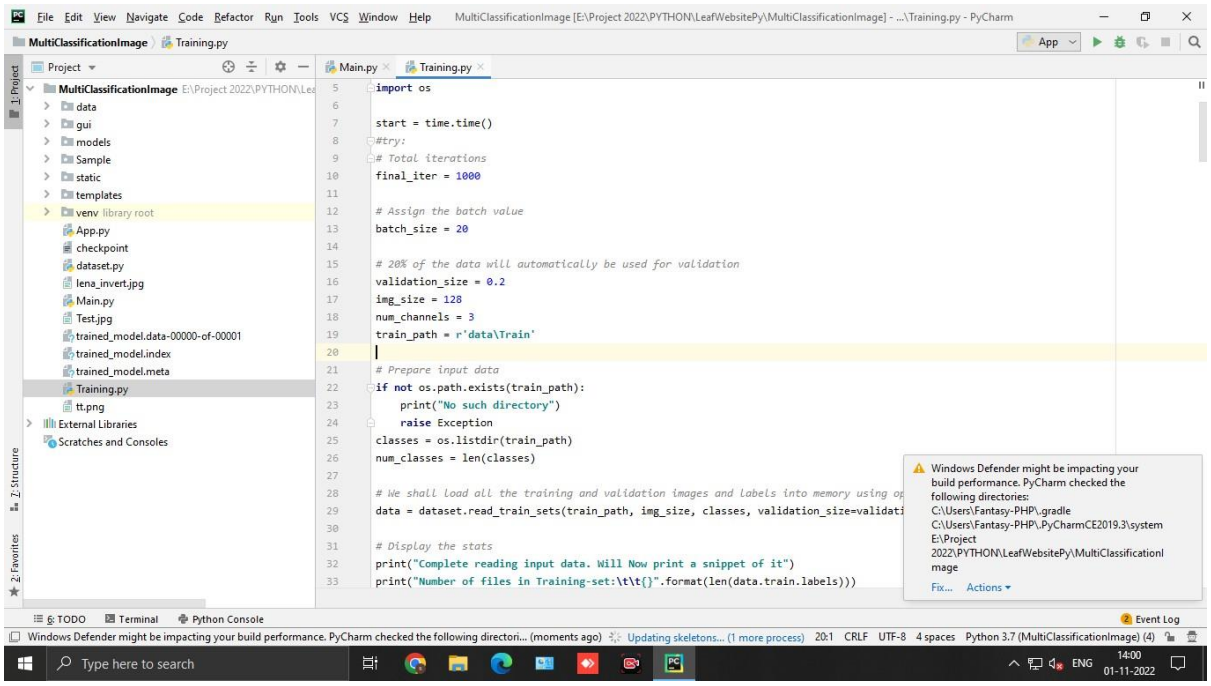
```
    filename = 'data/alien_test/Test.jpg'
cv2.imwrite(filename,  image)
print("After saving image:")
```

```
def main_account_screen():
from PIL import Image, ImageTk
global main_screen
    main_screen = Tk()
    width = 600
height = 600
screen_width = main_screen.winfo_screenwidth()
    screen_height = main_screen.winfo_screenheight()
    x = (screen_width / 2) - (width / 2)
    y = (screen_height / 2) - (height / 2)
    main_screen.geometry("%dx%d+%d+%d" % (width, height, x, y))
    main_screen.resizable(0, 0)
# main_screen.geometry("300x250")
main_screen.title("Leaf Disease classification")

    Label(text="Leaf Disease classification", bg="turquoise", width="300",
height="5", font=("Calibri", 16)).pack()
    Label(text="").pack()
    Label(text="").pack()

    image = ImageTk.PhotoImage(Image.open('gui/12344.jpg'))

    Label(main_screen, text='Hello', image=image, compound='left', height="100",
width="200",).pack()

    Button(text="Training", font=(
'Verdana', 15), height="2", width="30", command=training,
highlightcolor="black").pack(side=TOP)
    Label(text="").pack()
    Button(text="Testing", font=(
'Verdana', 15), height="2", width="30", command=testing).pack(side=TOP)

    Label(text="").pack()

    main_screen.mainloop()


main_account_screen()
```

APPENDIX -2

OUTPUT

SCREENSHOTS:



"E:\Project 2022\PYTHON\LeafWebsitePy\MultiClassificationImage\venv\Scripts\python.exe" "E:/Project 2022/PYTHON/LeafWebsitePy/MultiClassificationImage/Training.py"

Going to read training images

Now going to read Black spot files (Index: 0)

Now going to read canker files (Index: 1)

Now going to read greening files (Index: 2)

Now going to read healthy files (Index: 3)

Now going to read Melanose files (Index: 4)

Complete reading input data. Will Now print a snippet of it

Number of files in Training-set: 488

Number of files in Validation-set: 121

2022-11-01 14:01:22.019284: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2

WARNING:tensorflow:From E:/Project 2022/PYTHON/LeafWebsitePy/MultiClassificationImage/Training.py:40: calling argmax (from

tensorflow.python.ops.math_ops) with dimension is deprecated and will be removed in a future version.

Instructions for updating:

Use the `axis` argument instead

WARNING:tensorflow:From E:/Project 2022/PYTHON/LeafWebsitePy/MultiClassificationImage/Training.py:82: The name tf.nn.max_pool is deprecated. Please use tf.nn.max_pool2d instead.

Training Epoch 1/42 --- Training Accuracy: 45.0%, Validation Accuracy: 65.0%, Validation Loss: 1.396

Training Epoch 2/42 --- Training Accuracy: 20.0%, Validation Accuracy: 25.0%, Validation Loss: 1.316

Training Epoch 3/42 --- Training Accuracy: 20.0%, Validation Accuracy: 10.0%, Validation Loss: 1.259

Training Epoch 4/42 --- Training Accuracy: 30.0%, Validation Accuracy: 25.0%, Validation Loss: 1.241

Training Epoch 5/42 --- Training Accuracy: 35.0%, Validation Accuracy: 30.0%, Validation Loss: 1.229

Training Epoch 6/42 --- Training Accuracy: 40.0%, Validation Accuracy: 30.0%, Validation Loss: 1.207

Training Epoch 7/42 --- Training Accuracy: 40.0%, Validation Accuracy: 30.0%, Validation Loss: 1.191

Training Epoch 8/42 --- Training Accuracy: 40.0%, Validation Accuracy: 30.0%, Validation Loss: 1.211

Training Epoch 9/42 --- Training Accuracy: 40.0%, Validation Accuracy: 30.0%, Validation Loss: 1.198

Training Epoch 10/42 --- Training Accuracy: 40.0%, Validation Accuracy: 35.0%, Validation Loss: 1.148

Training Epoch 11/42 --- Training Accuracy: 40.0%, Validation Accuracy: 45.0%, Validation Loss: 1.107

Training Epoch 12/42 --- Training Accuracy: 40.0%, Validation Accuracy: 50.0%, Validation Loss: 1.065

Training Epoch 13/42 --- Training Accuracy: 40.0%, Validation Accuracy: 50.0%, Validation Loss: 1.041

Training Epoch 14/42 --- Training Accuracy: 50.0%, Validation Accuracy: 50.0%, Validation Loss: 0.996

Training Epoch 15/42 --- Training Accuracy: 55.0%, Validation Accuracy: 65.0%, Validation Loss: 0.931

Training Epoch 16/42 --- Training Accuracy: 55.0%, Validation Accuracy: 65.0%, Validation Loss: 0.880

Training Epoch 17/42 --- Training Accuracy: 55.0%, Validation Accuracy: 65.0%, Validation Loss: 0.849

Training Epoch 18/42 --- Training Accuracy: 55.0%, Validation Accuracy: 70.0%, Validation Loss: 0.811

Training Epoch 19/42 --- Training Accuracy: 65.0%, Validation Accuracy: 70.0%, Validation Loss: 0.787

Training Epoch 20/42 --- Training Accuracy: 65.0%, Validation Accuracy: 75.0%, Validation Loss: 0.762

Training Epoch 21/42 --- Training Accuracy: 65.0%, Validation Accuracy: 75.0%, Validation Loss: 0.741

Training Epoch 22/42 --- Training Accuracy: 65.0%, Validation Accuracy: 70.0%, Validation Loss: 0.718

Training Epoch 23/42 --- Training Accuracy: 65.0%, Validation Accuracy: 70.0%, Validation Loss: 0.698

Training Epoch 24/42 --- Training Accuracy: 70.0%, Validation Accuracy: 70.0%, Validation Loss: 0.679

Training Epoch 25/42 --- Training Accuracy: 70.0%, Validation Accuracy: 70.0%, Validation Loss: 0.655

Training Epoch 26/42 --- Training Accuracy: 70.0%, Validation Accuracy: 75.0%, Validation Loss: 0.635

Training Epoch 27/42 --- Training Accuracy: 75.0%, Validation Accuracy: 75.0%, Validation Loss: 0.607

Training Epoch 28/42 --- Training Accuracy: 75.0%, Validation Accuracy: 80.0%, Validation Loss: 0.586

Training Epoch 29/42 --- Training Accuracy: 80.0%, Validation Accuracy: 80.0%, Validation Loss: 0.560

Training Epoch 30/42 --- Training Accuracy: 85.0%, Validation Accuracy: 85.0%, Validation Loss: 0.532

Training Epoch 31/42 --- Training Accuracy: 85.0%, Validation Accuracy: 85.0%, Validation Loss: 0.515

Training Epoch 32/42 --- Training Accuracy: 90.0%, Validation Accuracy: 85.0%, Validation Loss: 0.507

Training Epoch 33/42 --- Training Accuracy: 90.0%, Validation Accuracy: 85.0%, Validation Loss: 0.494

Training Epoch 34/42 --- Training Accuracy: 90.0%, Validation Accuracy: 85.0%, Validation Loss: 0.478

Training Epoch 35/42 --- Training Accuracy: 90.0%, Validation Accuracy: 85.0%, Validation Loss: 0.471

Training Epoch 36/42 --- Training Accuracy: 90.0%, Validation Accuracy: 85.0%, Validation Loss: 0.452

Training Epoch 37/42 --- Training Accuracy: 90.0%, Validation Accuracy: 85.0%, Validation Loss: 0.453

Training Epoch 38/42 --- Training Accuracy: 90.0%, Validation Accuracy: 85.0%, Validation Loss: 0.433

Training Epoch 39/42 --- Training Accuracy: 90.0%, Validation Accuracy: 85.0%, Validation Loss: 0.437

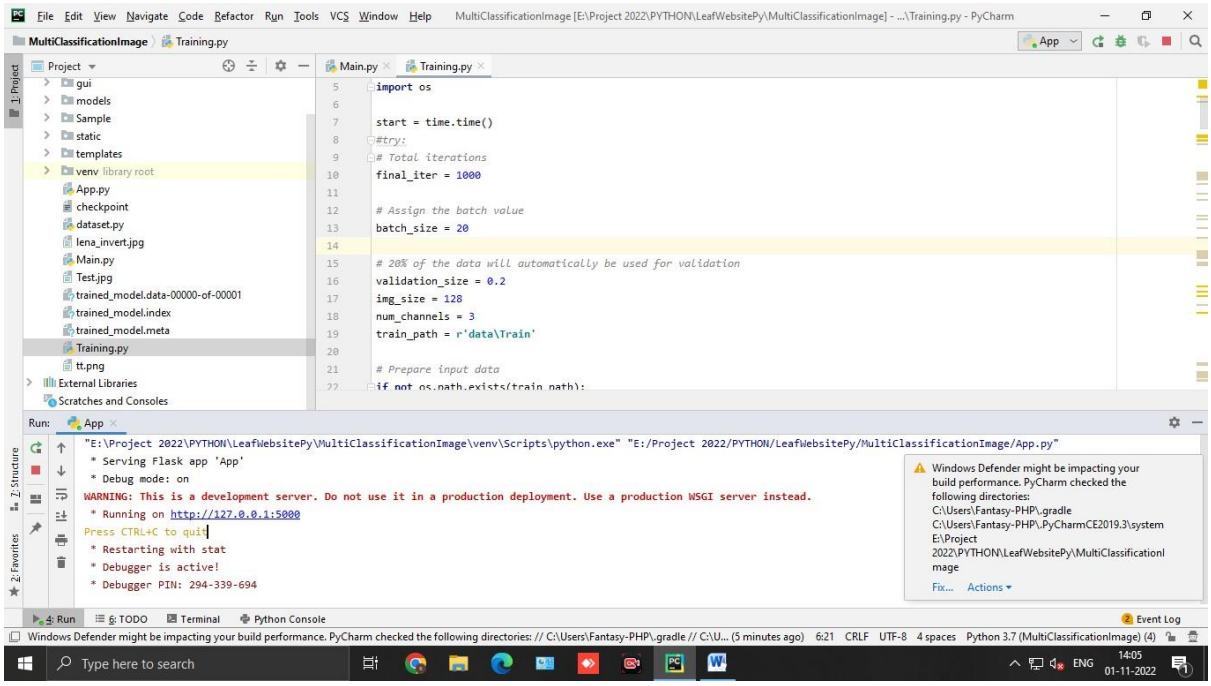
Training Epoch 40/42 --- Training Accuracy: 90.0%, Validation Accuracy: 85.0%, Validation Loss: 0.417

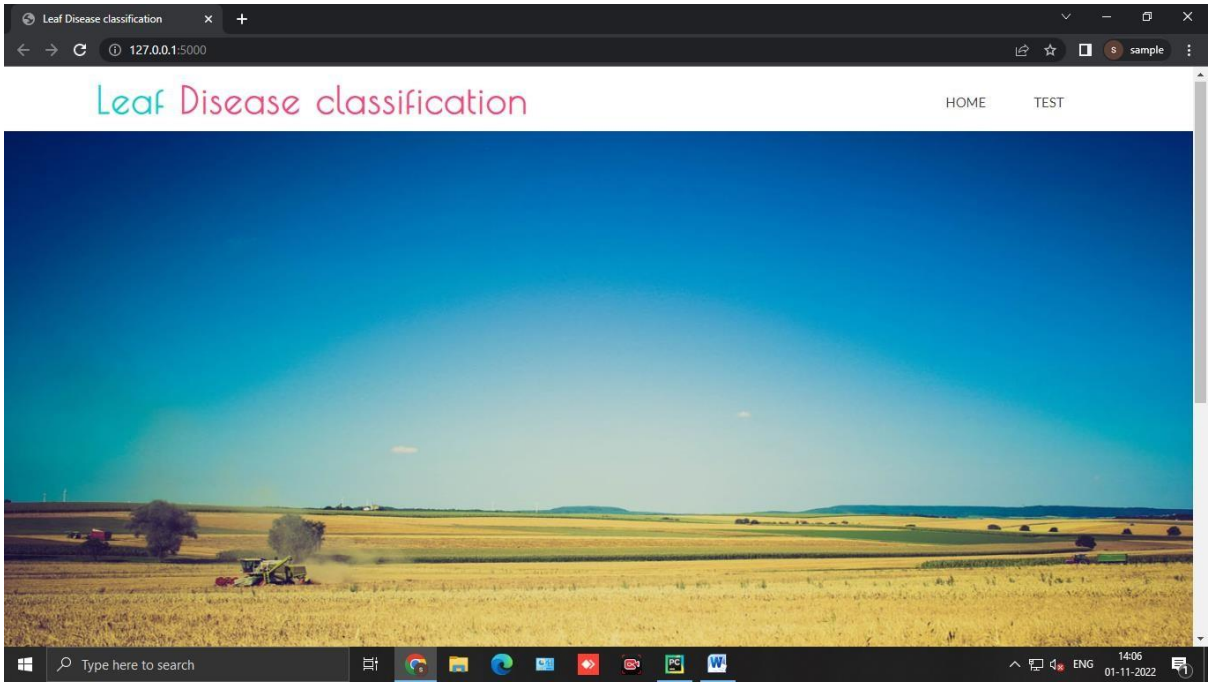
Training Epoch 41/42 --- Training Accuracy: 90.0%, Validation Accuracy: 85.0%, Validation Loss: 0.433

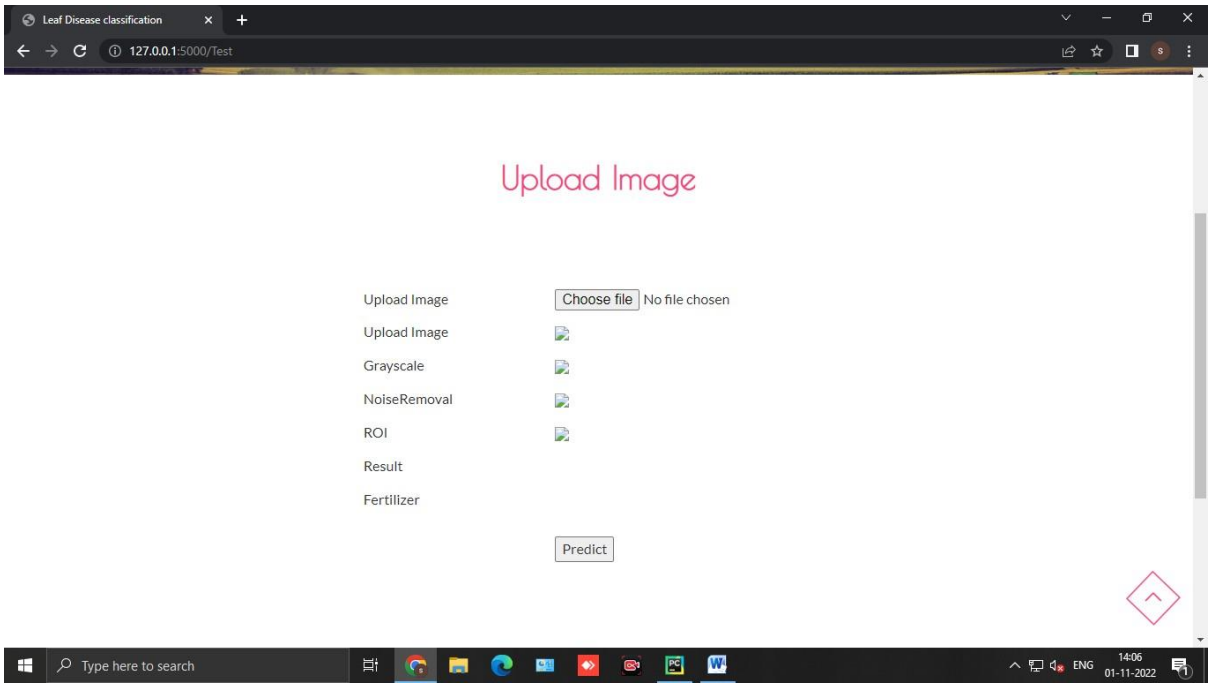
Training Epoch 42/42 --- Training Accuracy: 90.0%, Validation Accuracy: 85.0%, Validation Loss: 0.401

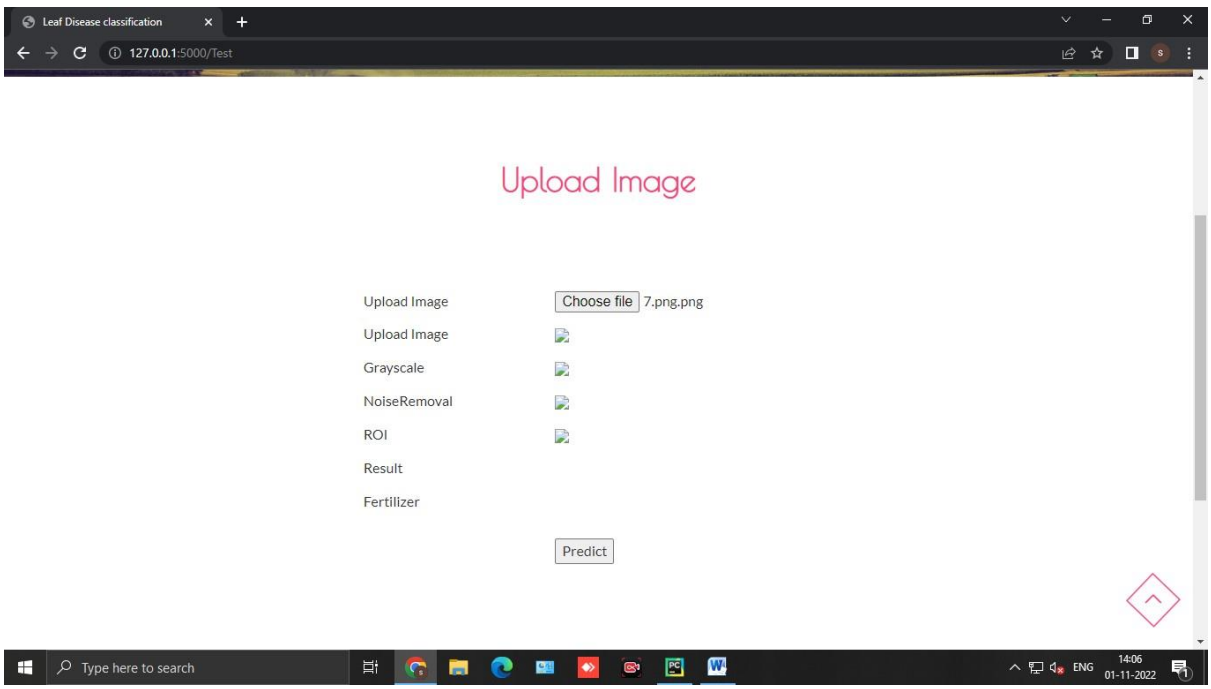
Execution Time: 3.9937183459599814 minutes

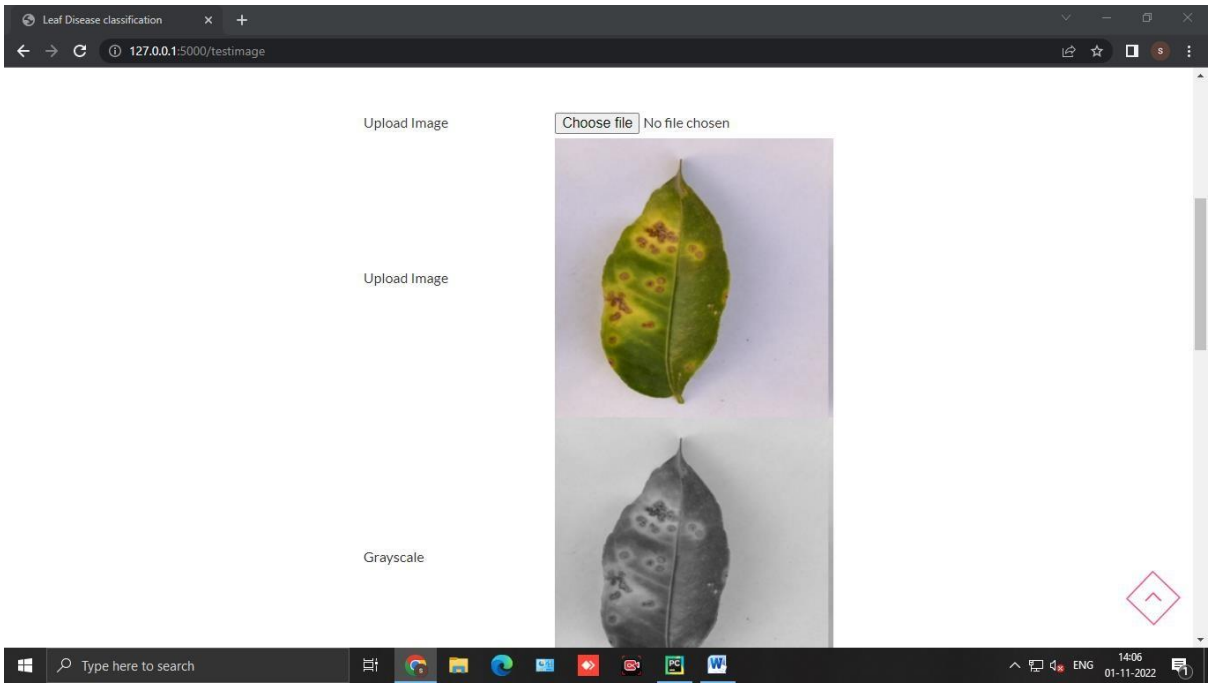
Process finished with exit code 0

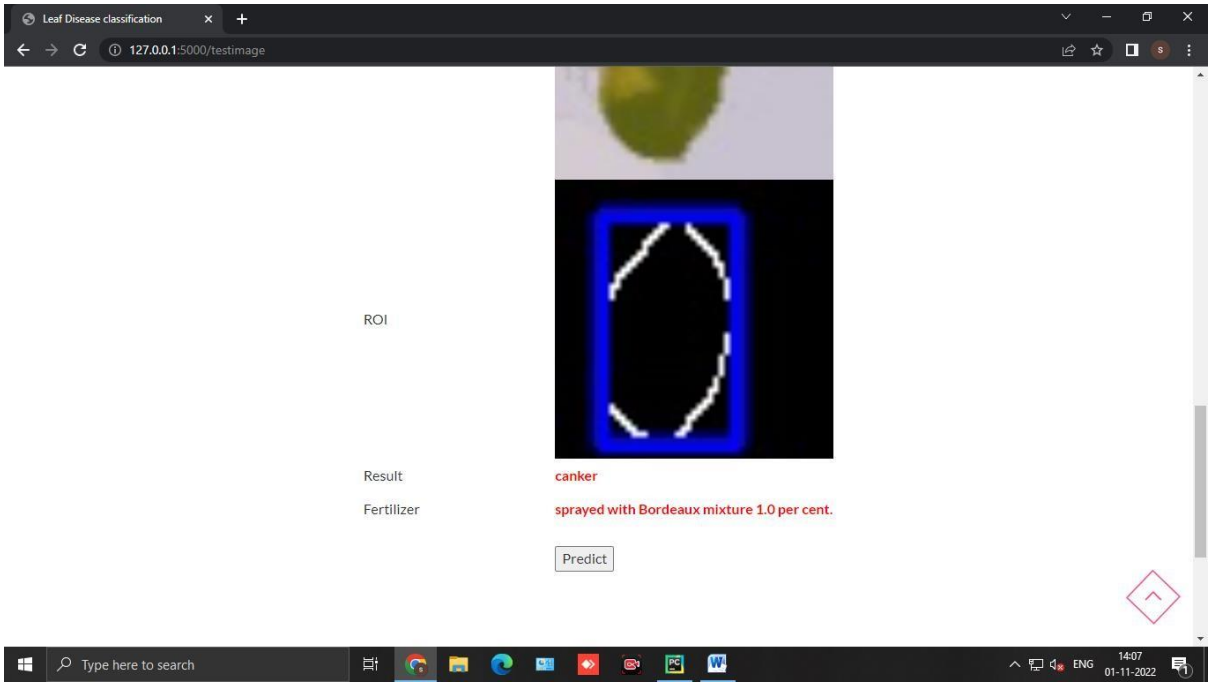

File Edit View Navigate Code Refactor Run Tools VCS Window Help
MultiClassificationImage [E:\Project 2022\PYTHON\LeafWebsitePy\MultiClassificationImage] - ...\Training.py - PyCharm
MultiClassificationImage
Training.py
App
Project
gui
models
Sample
static
templates
venv library root
App.py
checkpoint
dataset.py
lena_invert.jpg
Main.py
Test.jpg
trained_model.data-00000-of-00001
trained_model.index
trained_model.meta
Training.py
tt.png
External Libraries
Scratches and Consoles
Main.py
Training.py
import os
start = time.time()
#try:
# Total iterations
final_iter = 1000
# Assign the batch value
batch_size = 20
# 20% of the data will automatically be used for validation
validation_size = 0.2
img_size = 128
num_channels = 3
train_path = r'data\Train'
# Prepare input data
if not os.path.exists(train_path):
Run:
App
"E:\Project 2022\PYTHON\LeafWebsitePy\MultiClassificationImage\venv\Scripts\python.exe" "E:/Project 2022/PYTHON/LeafWebsitePy/MultiClassificationImage/App.py"
* Serving Flask app 'App'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 294-339-694
Windows Defender might be impacting your build performance. PyCharm checked the following directories:
C:\Users\Fantasy-PHP\.gradle
C:\Users\Fantasy-PHP\.PyCharmCE2019.3\system
E:\Project 2022\PYTHON\LeafWebsitePy\MultiClassificationImage
Fix... Actions
4: Run
6: TODO
Terminal
Python Console
Event Log
Windows Defender might be impacting your build performance. PyCharm checked the following directories: // C:\Users\Fantasy-PHP\.gradle // C:\U... (5 minutes ago) 6:21 CRLF UTF-8 4 spaces Python 3.7 (MultiClassificationImage) (4)
Type here to search
ENG
14:05
01-11-2022


Leaf Disease classification
127.0.0.1:5000
sample
Leaf Disease classification
HOME
TEST
Type here to search
ENG
14:06
01-11-2022


Leaf Disease classification
127.0.0.1:5000/Test
Upload Image
Upload Image
Choose file
No file chosen
Upload Image
Grayscale
NoiseRemoval
ROI
Result
Fertilizer
Predict
Type here to search
ENG
14:06
01-11-2022


Leaf Disease classification
127.0.0.1:5000/Test
Upload Image
Upload Image
Choose file
7.png.png
Upload Image
Grayscale
NoiseRemoval
ROI
Result
Fertilizer
Predict
Type here to search
ENG
14:06
01-11-2022


Leaf Disease classification
127.0.0.1:5000/testimage
Upload Image
Choose file
No file chosen
Upload Image
Grayscale
Type here to search
ENG
14:06
01-11-2022


Leaf Disease classification
127.0.0.1:5000/testimage
ROI
Result
canker
Fertilizer
sprayed with Bordeaux mixture 1.0 per cent.
Predict
Type here to search
ENG
14:07
01-11-2022