"AN INTEGRATED APPROACH FOR AUTOMATED EXTRACTION AND SEGMENTATION OF TOBACCO PLANT REGIONS USING CNN AND WATERSHED ALGORITHM"

[1]Prof. Rumana Anjum, [2]Abhishek S, [3]Harsha M, [4]Gayathri M, [5]J Sumanth

[2,3,4,5]Student
Computer Science and Engineering
Vidya Vikas Institute of Engineering and Technology
Mysore.

INTRODUCTION

When we utilize photographs or 2D paintings for representation purposes with regards to objects or people, we offer viewers with the chance to perceive their appearance through visual depictions in the form of images. Amplitude distribution is used in pictures processing as one way for determining their coloring effect.2D imagery is exemplified by various presentations such as photos, screens and even holograms - all created with three dimensions in mind. The figure displayed below highlights one clear case-in-point.

An image encompasses any two-dimensional form - from maps to graphs to pie charts and paintings. While some are crafted by hand through techniques such as drawing or painting; others are mechanised with the employment of printing or computer graphics technologies. In some instances however, these methods are combined for example when creating faux-photographs.

The visual realm can be broken down into two fundamental types: volatile and fixed images. Volatile images are those that disappear quickly after coming into existence – examples could be mirror reflections or cathode ray tube pictures. Conversely fixed images have been saved onto tangible mediums such as paper or cloth. Digital image processing involves using computer algorithms to manage photographic content – its a subset of digital signal processing within computer science and offers distinct benefits over analog methods due to its diverse range of algorithms available for managing input data. With these multiple algorithms issues like noise buildup and signal distortion are far less likely to occur.

IMAGING DEVICES

The creation of images through various imaging gadgets is an essential part of modern scientific research. The pioneers like Galileo Galilei and Newton aimed at designing & enhancing image technologies in their time. Today there are multiple imaging devices like

- eye-glasses
- camera obscura,
- magnifying glasses;
- microscope;
- telescope;
- holography available.

Features of Digital Imaging & Image Processing

Digital imaging systems have advanced significantly owing to the presence of computers & processors today making them highly versatile. One key advantage offered by digital computers over traditional analog electrical & optical information processing systems is the ability to reprogram without any hardware changes for undertaking new tasks. Simply developing a suitable computer code can enable you to create complex problem solvers using the same resources- hardware or software that were used earlier too. The swift adaptation capability towards different signal types as per user requirements makes computers an effective tool for

adaptive image processing. Signal convolution and correlation in analog optics involve transformations such as spatial and temporal Fourier analysis. Nonetheless, imaging systems that incorporate computers can perform any operation. Optoelectronic information processing is now extremely powerful when combined with digital signal processing.

Objectives:
The ability to identify and recognize specific objects or groups within images serves as an integral component across numerous applications.

Existing System:
The necessity for identifying objects or groups within photographs reaches varying industries such as agriculture and surveillance sectors. Developments have been made to automatically recognize photos containing members of an object class via research findings. It's worth noting that appropriate attention is critical when identifying the tobacco plant.

Proposed System:
In this study, we devised a three-phase system commencing with morphological procedures followed by watershed segmentation for potential tobacco plant regions via UAV photos. The creation and training of a deep convolutional neural network (CNN) serve as the subsequent step responsible for distinguishing between distinct types of tobacco plants while removing non-tobacco plants through post-processing measures.

Tobacco plant area extraction comprises four steps including
•noise filtering;
•soil region detection;
•plant region segmentation;
•and finally, plant region extraction.

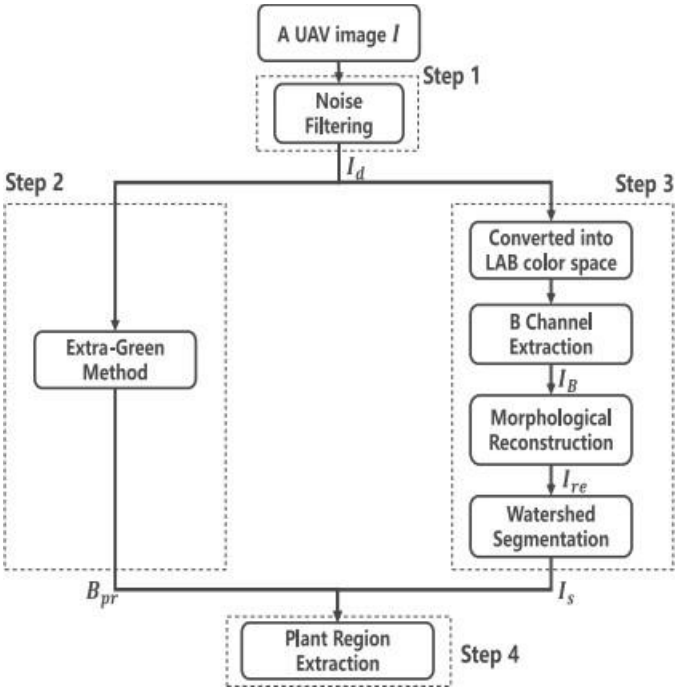


Fig1.2: Framework for Plant region extraction

Tobacco plants may be detected in UAV photos using a novel technique based on deep neural networks. According to our knowledge, this is the first study to look for tobacco plants in photographs captured by an unmanned aerial vehicle (UAV). A dataset of UAV images is used to test the effectiveness of the algorithm under consideration.

LITERATURE SURVEY
L. Meng and J. P. Kerekes, "Object tracking using high resolution satellite imagery,"[1]
There are several uses for high-resolution multispectral satellite photos that may be seen from multiple angles. A three-step system for monitoring moving objects, modeling targets, and finding matches for those targets is presented. The time-series photos are used to identify items that may be moving. Next, spectral and spatial information are extracted from the target to create a model. As part of the matching process, a new regional operator is created by combining the Bhattacharyya distance with the histogram intersection and pixel count similarity. The WorldView-2 satellite provided us with a series of multi-angle sequence photos, which we used to test our technique. The test's recall, precision, and F1 score are used to evaluate the tracking performance. Using high-resolution multispectral satellite images, we were able to show object tracking in a challenging setting in this research.

JIANG HUIXIAN, The Analysis of Plants Image Recognition Based on Deep Learning and Artificial Neural Network [2]
In order to better understand and protect the plants that we come into contact with on a daily basis, categorization and identification of plants is essential. When identifying a plant's identification, it is important to look at its leaves. Artificial intelligence and machine vision are being used to improve plant classification and protection using image analysis-based plant leaf identification technologies. "Deep learning" is the shorthand for neural networks at the heart of deep learning. Plant leaf samples are automatically trained and categorized using an artificial neural network that uses the back propagation approach. Image analysis will be used to extract leaf characteristics and identify plant species as the major objective of this project. Images of plant leaves are first segmented using a variety of techniques before being subjected to an algorithm for extracting the form and texture of individual leaves. Data on the chemical characteristics of plant leaves is then collected and organized in line withget all of the relevant information that is out there. KNN-based neighborhood classification, SVM-based support vector machine, and Kohonen network based on self-organizing feature mapping are used to investigate and compare 50 plant leaf datasets. In addition, ginkgo leaves were shown to be simpler to recognize from those of seven other kinds of trees.. Leaf photographs with a complex background have an excellent recognition effect. Using photos from the test set, the learning model calculates reconstruction errors. Recreating a deep learning model with the smallest number of errors will provide the test set's class label. In terms of recognition time, we discovered that this strategy is the quickest and most accurate.
Summary
This chapter focuses on the publications and resources that were consulted while writing this dissertation. All of these articles and websites discuss how to teach group conduct, as well as the many approaches that have been tried, as well as the benefits and drawbacks of each.

SYSTEM REQUIREMENT SPECIFICATION
The foundation of an effective software development process lies in its specifications that outline all critical elements and needs of the system. Software requirements specification (SRS), in particular, highlights a company's preliminary comprehension of their customer or prospect's demands and dependencies on paper. Both parties need to have an open line of communication for smooth collaboration throughout the project lifecycle while using SRS as a cost control mechanism. Besides acting as a primary source document for developers, SRS is also known as its parent document because it links other documents such as design specs, software architecture specs, testing plans, validation plans documentation plans and statements of work with ease. However, one must know that an SRS does not cover anything beyond what developers perceive their clients' system needs.
3.1Functional Requirement
When a system is provided with specified inputs or situations, a Functional Requirement specifies how the system should respond. Some examples of these are computations and data manipulation and processing. In this system, the functional requirements are as follows:-
1.Load the dataset into memory.
2.Load the picture from the input.
3.For segmentation, use Watershed.
4.Using CNN, train a dataset.
5.Using CNN, classify the picture.
3.2Non-functional Requirement
Requirements that do not directly affect the system's operation are known as non-functional requirements (NFR). A system's performance is judged by its ability to meet a set of predetermined criteria, rather than by its unique actions. It's possible that they have anything to do with emergent system characteristics like dependability, reaction speed, and store occupancy.. Non-functional requirements may be triggered by a variety of circumstances, including those related to the demands of the end user, budgetary restrictions, corporate regulations, and the desire for compatibility with other software and hardware systems.
1.Requirements for the product
2.Requirements of the Organization
3.A user's needs
4.Requirements for a basic operation
We have established a number of requirements to ensure our product meets user expectations:
Portability – Our program is built using Python ensuring its compatibility on any platform supporting said language.
Correctness – We pride ourselves in our meticulous approach utilizing well-defined criteria and methods resulting in accurate data output after rigorous testing.
Ease of Use – Our intuitive front-end interface facilitates seamless interaction without compromising functionality or productivity.
Modularity – Breaking down our software into smaller modules enhances flexibility while maintaining smooth interfacing via defined interfaces.
Robustness – Our aim is achieving peak performance while still optimizing speed without sacrificing accuracy. Python's robust attributes lower the system failure rate which belongs under non-functional needs further classified into execution quality or evolution quality subcategories.
Aspects like testing, maintenance, scalability or expandability are part of evolution quality which can be observed during development. University educated individuals possess a higher level of cognitive ability and analytical skills making them more capable of comprehending complex ideas and engaging in critical thinking. This is particularly important in todays society, where issues are becoming increasingly complex and multifaceted. As such having a university education is beneficial for both personal development and career success. Moreover research has consistently shown that university graduates earn higher salaries than those

without a degree. According to the US Census Bureau individuals with a bachelors degree earn on average $30,000 more per year than those with only a high school diploma. This disparity only widens as one climbs the educational ladder.

However it is not just about financial gain. University education also provides individuals with opportunities for personal growth and self discovery. Exposure to diverse perspectives and experiences can broaden ones worldview and lead to greater empathy and understanding towards others. In summary pursuing higher education has numerous benefits that extend beyond career advancement. It fosters critical thinking skills increases earning potential and promotes personal growth.

If you're into scientific programming with Python Anaconda has got you covered. It offers Spyder, an IDE designed specifically for this purpose. Not only that, but it also comes with an introspective and interactive testing environment thats part of the package. Once you've installed Anaconda simply run the command spyder to launch Spyder on Windows Mac OS X, or Linux. Alternatively if you prefer using Anaconda Navigator (also included with Anaconda) you can find Spyder on the Navigation Home tab there. To learn more about Spyder and its capabilities visit either the official Spyder website or consult the Spyder documentation. And speaking of command prompts: did you know that just like the regular command prompt Anacondas own version lets you run anaconda and conda commands without having to change folders or modify your path? This makes executing commands and scripts a breeze.

MATLAB® on the other hand is a programming language that directly expresses matrix and array mathematics within its desktop environment. This makes it ideal for iterative analysis or design processes. Seamlessly integrate code, output, and formatted text into one notebook with the help of the Live Editor.

Our MATLAB toolboxes are expertly crafted, meticulously tested, and extensively documented to guarantee superior quality. Discover various algorithms through our interactive apps that enable you to explore data interactions. Boost productivity by automating or repeating tasks using MATLAB's programming capabilities upon achieving optimal outcomes.

Python Programming

As far as general purpose high level programming languages go few surpass Pythons popularity and effectiveness. Designed by Guido van Rossum in 1991 with the goal of achieving superior code readability and minimalism in syntax so that programmers could successfully communicate their ideas in fewer lines of code.

1) Finding An Interpreter:

The first step towards implementing successful Python coding is finding an interpreter to execute your programs.

Fortunately many online interpreters are available for immediate use without any prior downloads.

Alternatively Windows users have free access to several interpreters, such as the native IDLE (Integrated Development Environment).

Chapter 4

SYSTEM ANALYSIS

To tackle complex challenges effectively its essential to take an analytical approach that considers various factors thoroughly. System analysis involves scrutinizing current problems in detail while also identifying objectives and outlining potential solutions using both technical tools and critical thinking skills.

One fundamental component in this process is the feasibility study—an assessment that outlines specific goals for design and development while also testing workability, resource efficiency, impact on organizational performance among other factors. Our team has started conducting extensive research to complete a comprehensive feasibility assessment based on prior research findings regarding our current issue at hand.

The eight crucial stages involved in carrying out an effective feasibility analysis are:

1) Establishing a project team with assigned leaders

2)Identifying potential systems proposed for implementation

3) Defining unique goals/features required from each potential system

4)Evaluating performance capabilities/cost effectiveness for every prospective solution Feasibility analysis serves as a critical starting point when considering any new project or product development. It involves examining three primary areas: economic feasibility (performance vs cost) technical feasibility (adaptability & implementation) & social acceptability (public opinion). Economic viability forms an integral part of considerations during this analysis phase, where we must carefully assess various performance metrics and associated costs related to each proposed weight system.

By comparing all available options thoroughly and selecting the most suitable one for our projects goals and resource allocation strategies- we can ensure maximum optimization of resources used while also meeting our goals effectively. With a final system selection made after extensively evaluating all options available under consideration during feasibility analysis - it is time now for us to prepare an informative report. Our report on selected recommendations for weight system implementation will provide key decision making information necessary for management approval.

4.1.1ECONOMICAL FEASIBILITY

This study is carried out to determine how the system will effect the company's bottom line. The company has a restricted budget for research and development of the system. The justification for expenditures must be provided. Because the majority of the technologies used are freely available, the constructed system was able to remain below budget. Only customized products were required to be purchased.

4.1.2TECHNICAL FEASIBILITY

The goal of this study is to look at the technological viability of the system, as well as its technical requirements. Any new system must not need a big number of technical resources to function. As a consequence, current technical resources will under significant strain. This will put the buyer under a lot of stress. It is essential to make little or no changes to the developed system in order to implement this system.

4.1.3SOCIAL FEASIBILITY

The study's purpose is to determine how effectively the system is accepted by its target audience. As part of this, we train the user how to utilize the technology most effectively. The user should not be terrified of the system, but rather accept it as a need. Only the methods employed to educate and acquaint the user with the system have any effect on how well he adopts the system. Because he is the system's last user, it is critical that his self-esteem be boosted so that he can offer some useful feedback.

Summary

The primary goal of this chapter is to determine whether or not the system is practical. So several types of analysis, such as performance and technical, economic and financial analysis are carried out for these reasons

Chapter 5

SYSTEM DESIGN

A well-designed system is one that is easy to use and understand. For the sake of specifying a procedure or system in sufficient depth to allow for its physical embodiment, "Design" is described as "The act of applying numerous approaches and concepts." The system is built using a variety of design principles. System features, components, and parts are all described in detail in the design specification for the benefit of end users.

5.1Fundamental Design Concepts

Three decades ago, a set of core design ideals began to emerge. No matter how popular a certain idea may be at any one moment, it has stood the test of time. One may build upon the other to create a more complex design process from a solid base. In order to "get it right," designers rely on a set of core design ideas. There are several essential design ideas that are used in this project to ensure that the specifications are met.

5.1.1Input Design

User-oriented inputs are transformed into computer format via the process of input design. Input data should be designed in such a way that it makes automation as simple and error-free as feasible. The application's user interface is designed to make it simple for users to enter and choose data. When developing a project, consideration is given to input design needs such as user friendliness, consistency in format and interaction with users to ensure that they get exactly what they need and when they need it at all times. It is essential that the input design be given the utmost care since it is an integral component of overall system design. One of the most costly parts of a system is gathering input data since this data has to be routed via several modules. As a result of this, if a user's IP address is unknown when they're ready to transmit the data to the destination, it might lead to errors.

5.1.2Design of Output

A high-quality output meets the demands of the end user while also successfully communicating the information. In every system, outputs are utilized to communicate the results of processing to the user and other systems. It provides the user with the most relevant and up-to- date information possible. The system's interaction with the source and destination machines is improved through efficient and intelligent output. Computer outputs are needed largely to ensure that the user receives the identical packet that they sent, rather than damaged or faked packets. These findings are also utilized to save a copy of them for future reference.

5.1.3The MVC Design Method

The model-delegate is a simplified version of the MVC concept that is used in Swing. In this architecture, both the view and the controller object are combined into a single element called as the UI delegate. It is now possible to converse back and forth between the model and the user interface delegate. Model and UI delegate are the two components that make up a Swing component. The model is in charge of keeping track of the current condition of each component. Each component's UI delegate is responsible for ensuring that the necessary information is available for how the component should be shown on the computer screen. A variety of events propagate across the component, and the UI delegate responds to them (along with AWT).

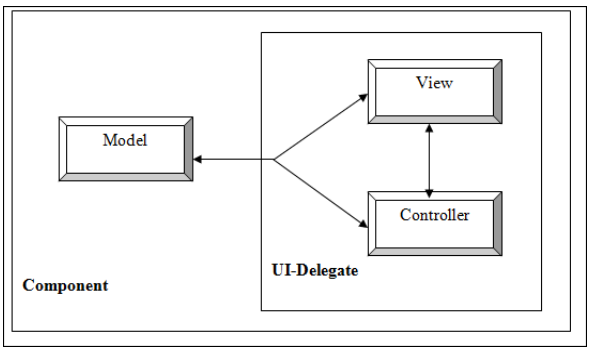


Fig 5.1- View and Controller are combined to form a UI delegate object.

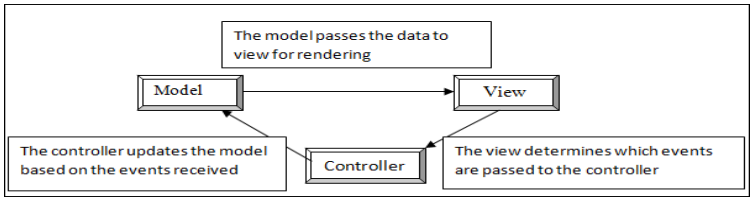


Fig 5.2- The MVC design allows for communication.

The architecture of the system was designed using the MVC design pattern. As the foundation for each of its components, the model-view-controller (MVC) architecture is used by the Swing framework. A GUI component is broken into three parts by MVC. The component's behavior is heavily influenced by each of these factors. A model, a view, and a controller are all parts of an MVC-style software application.

Model

The component's state and low-level behaviour are represented by the model. It's in charge of maintaining the current state and carrying out any changes on it. Neither the controllers nor the views are known to the model. Each component's status data is included. There are a variety of models to choose from depending on the sort of component you're dealing with. Information about the current location of the "thumb," its minimum and maximum values and its width might be included in a scrollbar component's model. A menu, on the other hand, may only show the options from which the user may choose. When the model's state changes, the system informs the views and keeps track of the linkages between the model and the views.

View

An element's "view" describes how you see it displayed on the screen. The model's visual representation is handled by this component. A title bar usually runs the full height of the window frame in the vast majority of cases. Close boxes may be on either the left or right side of the title bar. The following are instances of distinct views for the same window object.. Models may have several views, however in the Swing set, this is uncommon.

Controller

The model's user interface is controlled by the controller. The model's state may be modified using this approach. A component's interaction with events is governed by its user interface.

The view will not be able to properly draw the scrollbar until it first obtains data from the model. Unless the scrollbar can determine its current location and width in relation to the minimum and maximum values, the "thumb" cannot be shown. It is determined by the view as to whether or not user events, including mouse clicks are received by components. It's the controller's decision whether or not to act on these occurrences. The values in the model may have to be changed based on the controller's choice. As a result of scrolling, the user's thumb position in the model will be incremented. The cycle may be repeated at this point.

It is possible to divide the JFC user interface into three parts: the model, the view, and the controller. The basic MVC paradigm is often adapted by combining the view and controller into a single unit. They're responsible for the component's user-interface design.

5.2 Methodology for System Development

5.3 Methodology for System Development

In order to produce a product or solve an issue, the system development technique must be followed. There are many stages, techniques, and actions involved in creating a piece of software. For product development, it follows a set of stages. The waterfall model is used in this project's development.

5.3.1 Model phases

Requirement Analysis: This phase focuses on gathering the system's requirements. Documentation and requirements review are part of this step.

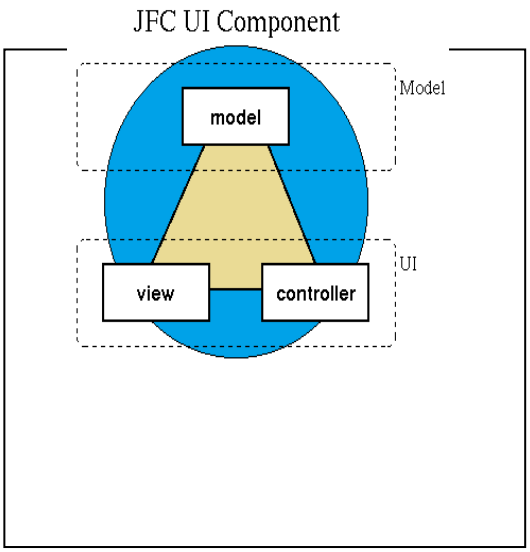


Figure 5.3- JFC user interface component.

System Design: The system specs are converted into a software representation with the needs in mind. Algorithms, data structures, and software architecture are some of the topics covered in this phase.

Coding: In this phase, the programmer begins coding in order to sketch out the product's features in complete detail. Machine-readable code is everything that is generated from system requirements.

Implementation: Software coding and programming are both part of the implementation process. Typically, the product of this phase includes the library, executables, user guides, and other documentation for the program.

Testing: In this step, each program (model) is put together and tested to verify that the software requirements are met. Verification and validation are the primary goals of testing.
Maintenance: Customers' needs vary over time; external environments change; faults and oversights discovered during testing are fixed; and software efficiency is improved throughout the maintenance phase of development.

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

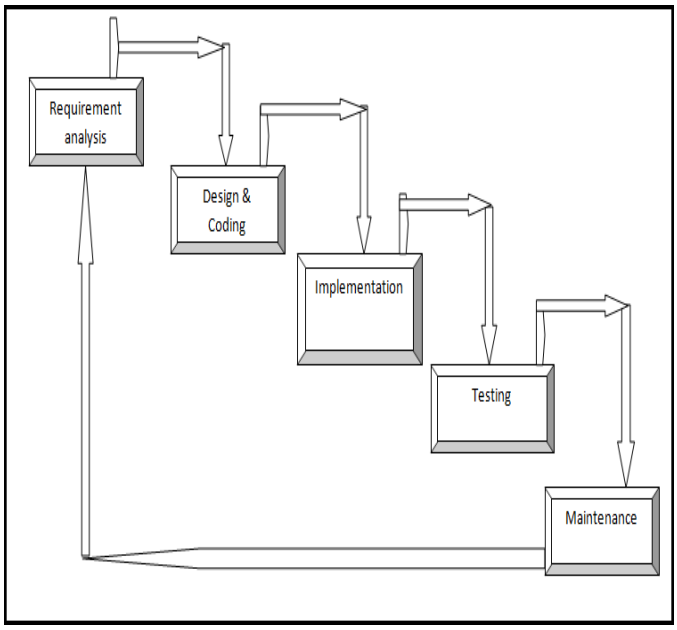
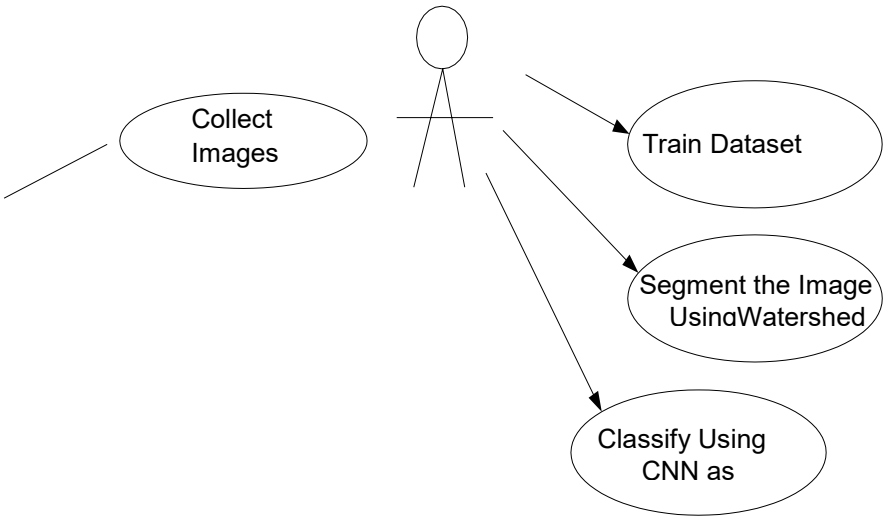


Fig 5.4:- The waterfall model

Architecture of the System

Conceptual design for a system is known as system architecture, which specifies the structure and behavior of the system itself. A formal description of a system, arranged to facilitate reasoning about the structural aspects of the system, is known as an architectural description. System components or building blocks are defined and a strategy is provided for procuring and developing items and systems that operate together to achieve the overall system.

.



The System architecture is shown below.

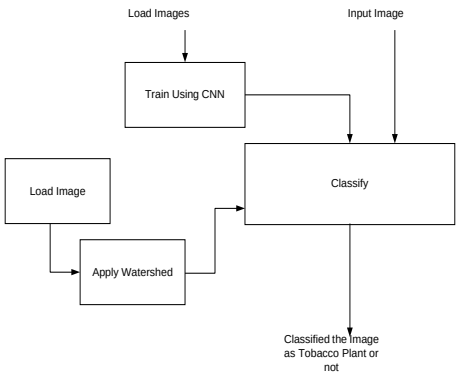
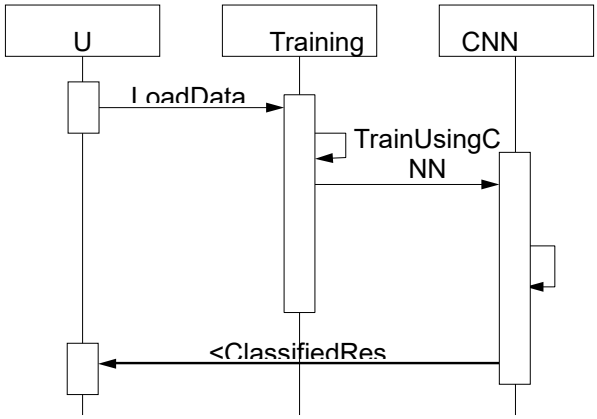


Fig 5.5: The System Architecture

Classes created specifically for the system

Using the Unified Modeling Language (UML), a class diagram is a form of static structural diagram that explains the structure of a system by depicting the system's classes, their properties, and their interactions. Below is a diagram of the classes.



The system's use case diagram

Diagrams developed from a use case study are called use case diagrams. Use cases, actors, and any relationships between them are all depicted graphically to provide a clear picture of the system's capabilities

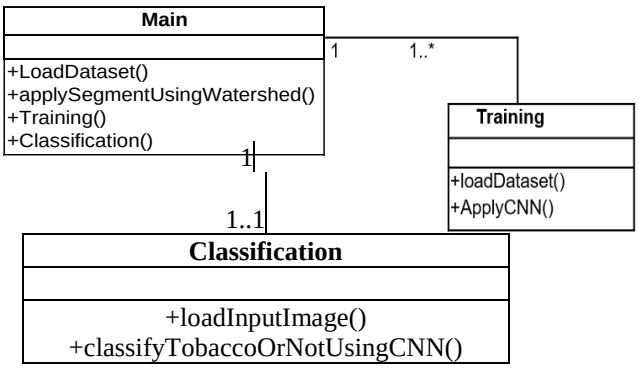


Fig 5.7:- The Use Case Diagram

Sequence diagram of system operation

Diagrams that indicate how processes interact with one another and in what order are known as sequence diagrams in the Unified Modelling Language (UML). It's a chart's structure. The diagrams are shown in the following order.

Segment

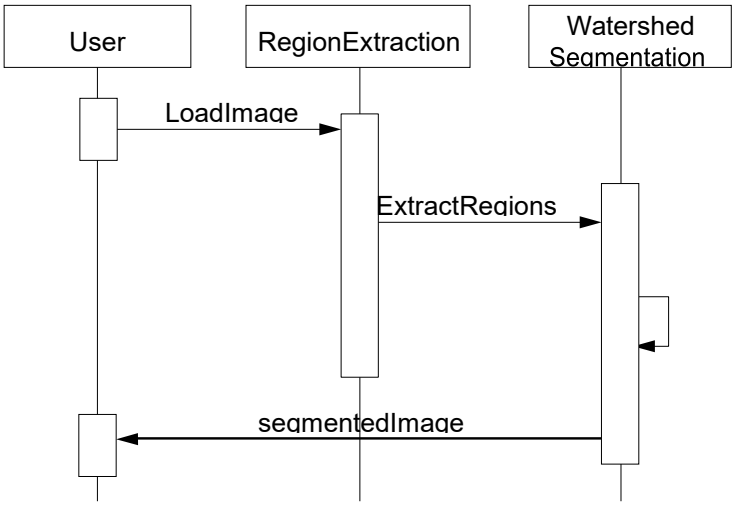


Fig 5.8- The Sequence Diagram

6.2 Methodology

1. Load the image
2. To aid the thresholding step

a. Apply pyramid mean shift filtering to help the accuracy of our thresholding step, and
b. Finally display our image

3. Then convert the image to grayscale and apply Otsu's thresholding to segment the background from the foreground.
4. Finally, the last step is to detect contours in the thresholder image and draw each individual contour:

CNN Algorithm
Step 1: Choose a Dataset
Step 2: Prepare Dataset for Training Step 3: Create Training Data
Step 4: Assigning Labels and Features Step 5: Split X and Y for use in CNN
Step 6: Define, compile and train the CNN Model.

Chapter 7:

Testing

Testing plays a crucial role in thoroughly evaluating a PC-based framework. Testers create a range of tests that simulate diverse scenarios, each with a specific purpose. However, the overall objective remains the same: ensuring that all components of the framework work together as intended and can achieve their stated goals. The item under test undergoes comprehensive testing to ensure it performs precisely as expected. Testing serves as the final stage in the association's testing and approval process.

The testing phase aims to accomplish the following objectives:

Reaffirming High Quality: The testing process seeks to validate the project's overall quality, ensuring that it meets the required standards.

Identifying and Correcting Issues: Any lingering issues from the previous steps are identified and addressed during testing. This phase acts as an opportunity to uncover and rectify any bugs, errors, or shortcomings within the framework.

Demonstrating Viability: Testing is conducted to demonstrate that the program is a viable solution to the original problem or challenge it was designed to address. By verifying its effectiveness, the testing phase validates the framework's suitability for its intended purpose.

Guaranteeing Smooth Functioning: The ultimate goal of testing is to ensure the smooth functioning of the entire system. This encompasses verifying that all components work together seamlessly and that the framework operates without any critical glitches or malfunctions.

While testing, the primary focus is directed towards examining and making necessary changes to the source code. This process involves analyzing the underlying programming instructions to identify any areas that require improvement or correction. By scrutinizing the source code, testers can ensure the reliability and functionality of the framework's programming logic.

6.4 Implementation

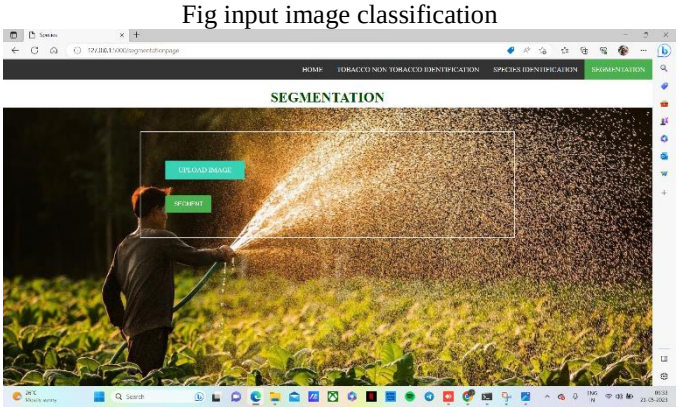
The following images depict the outputs or outcomes we may expect after running the

system's various modules one by one.

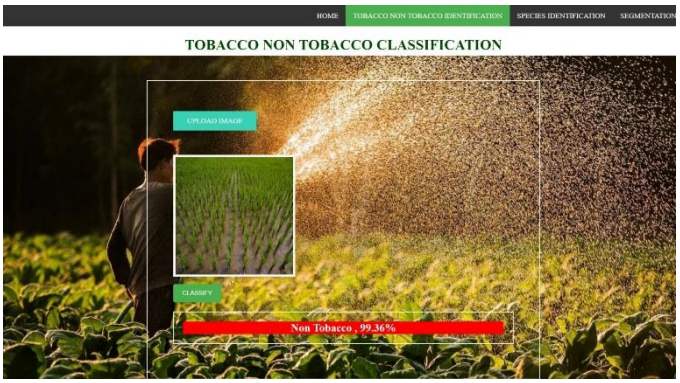
Fig 6.4.1: Frontend Design

Tobacco and Non-Tobacco Identification Front Page

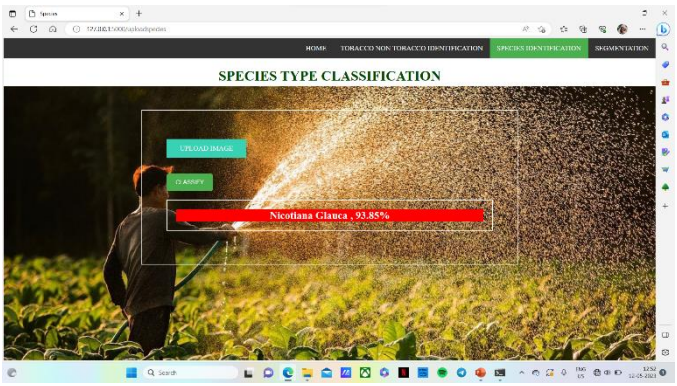
Fig input image classification



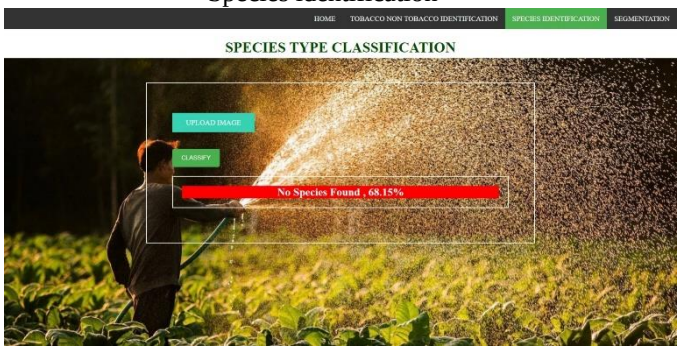
segmentation



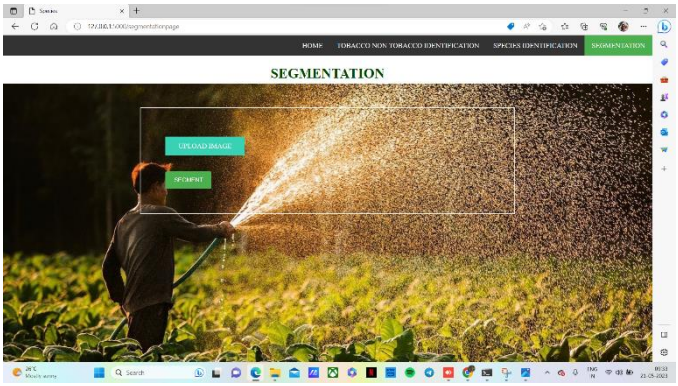
Non tobacco classification

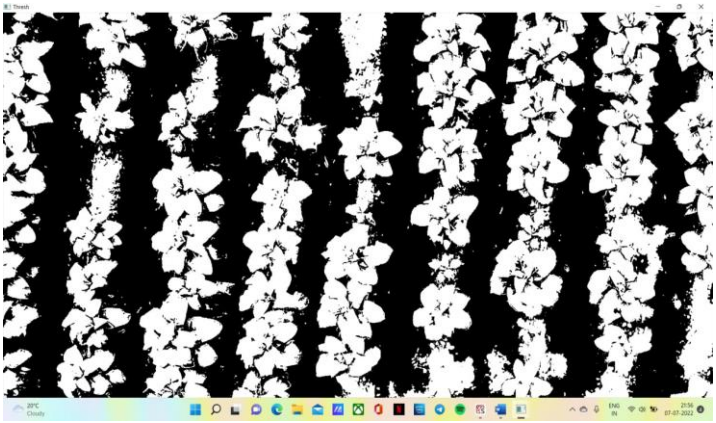
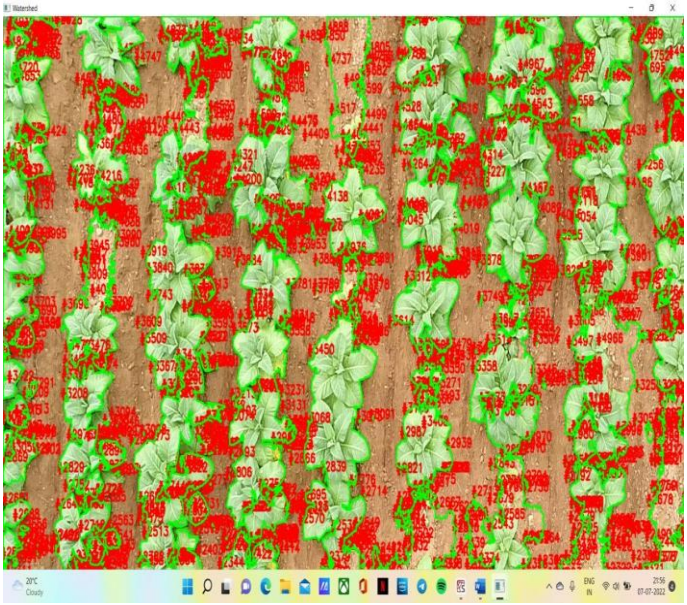


Species identification



No species found if it's not species of tobacco plant
Tobacco input image



Segment of Tobacco

tobacco input image