GROUP ID: GD12

A PROJECT REPORT ON

"TREE COUNTING AND DETECTION AUTOMATION USING CNN"

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE
IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)

SUBMITTED BY

Kush Ise	BECOD424
Abhiraj Deshpande	BECOD412
Roshan Bahiram	BECOD464
Kunwar Satyam Singh	BECOD431

DEPARTMENT OF COMPUTER ENGINEERING

PCET'S PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

Sector No. 26, Pradhikaran, Nigdi, Pimpri-Chinchwad, PUNE 411044
SAVITRIBAI PHULE PUNE UNIVERSITY

CERTIFICATE

This is to certify that the project report entitles

"Tree Counting and Detection Automation Using CNN"

Submitted by

Kush Ise	BECOD424
Abhiraj Despande	BECOD412
Roshan Bahiram	BECOD464
Satyam Singh	BECOD431

is a bonafide student of this institute and the work has been carried out by him/her under the supervision of **Prof. A. B. C** and it is approved for the partial fulfillment of the requirement of SavitribaiPhule Pune University, for the award of the degree of **Bachelor of Engineering** (Computer Engineering).

(Prof. Ganesh Kadam)
Guide
Department of Computer Engineering

(Prof. Dr. K. Rajeswari)
Head,
Department of Computer Engineering

(Prof. Dr. G.N. Kulkarni)
Director,
PimpriChinchwad College of Engineering Pune – 44

Place : Pune
Date :

ACKNOWLEDGEMENT

We express our sincere thanks to our **Guide Prof. Ganesh Kadam** for his/her constant encouragement and support throughout our project, especially for the useful suggestions given during the course of project and having laid down the foundation for the success of this work.

We would also like to thank our **Project Coordinator, Prof. Sushma R. Vispute**for her assistance, genuine support and guidance from early stages of the project. We would like to thank **Prof. Dr. K. Rajeswari, Head of Computer Department** for her unwavering support during the entire course of this project work. We are very grateful to our **Director, Prof. Dr. G.N. Kulkarni** for providing us with an environment to complete our project successfully. We also thank all the staff members of our college and technicians for their help in making this project a success.

We also thank all the web committees for enriching us with their immense knowledge. Finally, we take this opportunity to extend our deep appreciation to our family and friends, for all that they meant to us during the crucial times of the completion of our project.

NAME OF THE STUDENTS

Kush Ise

Abhiraj Deshpande

Roshan Bahiram

Satyam Singh

ABSTRACT

Palm tree is one of the economically grown crops in India. In this paper, we develop an automatic method for counting the number of coconut trees in UAV images. The availability of high resolution remote sensing images helps people in having large amounts of detailed digital imaging of vegetation areas. Today, high resolution drone images can be used to quickly estimate the number of coconut trees with little expense or labor. The purpose is to find innovative methods to determine coconut trees utilizing remote sensing. Deep learning techniques with convolutional neural network (CNN) algorithms is used to detect the coconut trees.

INDEX

LIST OF ABBREVATIONS

ABBREVIATION	ILLUSTRATION
VPN	Virtual Private Network
IP	Internet Protocol
IDS	Intrusion Detection System
TCP	Transmission Control Protocol

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW

The assignment of computerized palm tree detection is also similarly complicated with the aid of the truth that palm tree plantations range wildly in density and spatial association. The foundation for further evaluation work, such as yield prediction and fertilizer budget estimation, is localization. Palm tree localization the usage of photographs is not a broadly- studied trouble. The traditional method is to set up workers to plantations to manually remember trees . Capturing aerial pictures is demanding and also complexity of crop and field management. The above challenges mention above are a greater concern in the areas of agriculture field because of the presence of the so many trees and inefficient humanly tree counting one by one .

Challenges are difficult when we have to locate palm trees on the surface of the earth. This makes the work of administration department very difficult gives inadequate tree data for the advanced experts in agriculture. For this reason, They give an advanced amd sophisticated framework which helps to make a stock of crowns of the trees through automation of counting and geolocating them using aerial color photographs .

1.2 MOTIVATION

Tree count management is important for sustaining conservational stability and ecological biodiversity. A systematic tree inventory of the forested areas and in the urban areas can help us involvedly view the causes of decline of forests in the area, decline in green cover in urban areas etc. Challenges are difficult when we have to locate palm trees on the surface of the earth. This makes the work of administration department very difficult gives inadequate tree data for the advanced experts in agriculture. For this reason, They give an advanced and sophisticated framework which helps to make a stock of crowns of the trees through automation of counting and geolocating them using aerial color photograph.

1.3 PROBLEM STATEMENT

Counting tree using satellite 2D images and to develop a unsupervised learning system for counting and localizing trees in high-resolution imagery. To develop tree counting software using CNN .

1.4 PROJECT SCOPE AND LIMITATIONS

Future work may involve using multispectral images to locate the trees and determine which tree it is. Challenges are difficult when we have to locate palm trees on the surface of the earth. This makes the work of administration department very difficult gives inadequate tree data for the advanced experts in agriculture. For this reason, they give an advanced and sophisticated framework which helps to make a stock of crowns of the trees through automation of counting and geolocating them using aerial color photograph

CHAPTER 2
LITERATURE SURVEY

Paper 1: Coconut Trees Detection on the Tenarunga Using High-Resolution Satellite Images and Deep Learning

Author : Juepeng Zheng; Wenzhao Wu

Description : The Coconut tree is of great importance in economic values and ecological impacts for many tropical developing countries and lots of islands in the Pacific Ocean. Detecting and counting coconut is a meaningful and valuable research. In this paper, we present a coconut tree crown detection method to detect and count the coconut trees in the Tenarunga from high-resolution satellite images acquired by Google Earth. Our coconut tree detection method contains three major procedures: feature extraction, a multi-level Region Proposal Network (RPN) and a large-scale coconut tree detection workflow. We manually annotate all coconut trees for our study regions in the Tenarunga. Eventually, we achieve a higher average F1-score of 77.14% in our four test regions than pure Faster R-CNN. Experiment results demonstrate the potential for large-scale individual coconut tree detection and counting from high-resolution satellite images using deep learning.

Paper 2: Objcount an Evolving Spectral and Spatial Approach for Tree Count using Multispectral Satellite Images

Author : Akanksha Dokania; Neelanshi Varia

Description : In this paper, we present objcount framework for counting trees in multispectral satellite images, based on evolving spectral and spatial approach. Initially, Calinski-Harabasz criterion was used to determine the optimal number of clusters for the images. Unsupervised Extreme Learning Machine (US-ELM) was used for detecting the trees in the image and a counting algorithm based on distance transformation and spatial clustering was used for counting the number of detected trees. From the obtained result, we compare the performances of the proposed method on three multispectral satellite images with the state-of-the-art namely, K-means, Mean Shift Clustering (MSC) and Self-Organizing Maps (SOM). It was observed that US-ELM performed better than other methods.

Paper 3: Palm Trees Counting in Remote Sensing Imagery Using Regression Convolutional Neural Network

Author : KhelifaD jerriri ; Mohamed Ghabi ; Moussa SojianeKaroui

Description : Date palm trees are important economic crops in many countries and counting their numbers in a plantation area is crucial information for predicting the yield of date fruits, determination of insurance and financial aids, etc. In this abstract, a supervised tree counting framework is proposed using Convolutional Neural Network (CNN). The proposed approach casts the counting process into a regression problem, instead of following the classification or detection framework. To further decrease the prediction error of counting, we fine-tuned a pretrained CNN architecture into regression model. As the final output, not only the tree count is estimated for an image, but also its spatial density map is provided. Trained with small image patches cropped from airborne dataset, the proposed method is compared to manual counting and obtains good performance.

Paper 4 : Remote Sensing: An Automated Methodology for Olive Tree Detection and Counting in Satellite Images

Author: Umair Khan; Muhammad Waleed; Ashfaq Khan

Description: Cultivation of olive trees for the past few years has been widely spread across Mediterranean countries, including Spain, Greece, Italy, France, and Turkey. Among these countries, Spain is listed as the largest olive producing country with almost 45% of olive oil production per year. Dedicating land of over 2.4 million hectares for the olive cultivation, Spain is among the leading distributors of olives throughout the world. Due to its high significance in the country's economy, the crop yield must be recorded. Manual collection of data over such expanded fields is humanly infeasible. Remote collection of such information can be made possible through the utilization of satellite imagery. This paper presents an automated olive tree counting method based on image processing of satellite imagery. The images are pre-processed using the unsharp masking followed by improved multi-level thresholding-based segmentation. Resulting circular blobs are detected through the circular Hough transform for identification. Validation has been performed by evaluating the proposed scheme for the dataset formed by acquiring images through the “El Sistema de InformaciónGeográfica de ParcelasAgrícolas” viewer over the region of Spain. The proposed algorithm achieves an accuracy of 96% in

detection. Computation time was recorded as 24 ms for an image size of 300 × 300 pixels. The less spectral information is used in our proposed methodology resulting in a competitive accuracy with low computational cost in comparison to the state-of-the-art technique.

Paper 5: CNN Based Technique for Automatic Tree Counting Using Very High Resolution Data
Author : Aparna P.; Ramachandra Hebbar; Harshita M.P.
Description : Coconut is one of the economically grown crops in India. In this paper, we develop an automatic method for counting the number of coconut trees in UAV images. The availability of high resolution remote sensing images helps people in having large amounts of detailed digital imaging of vegetation areas. Today, the estimated coconut tree count can be determined in a short duration of time through high resolution drone images with low cost and labor. The goal is to find new methods to determine coconut trees using remote sensing. Deep learning techniques with convolutional neural network (CNN) algorithms is used to detect the coconut trees.

Paper 6: Toddy Palm Trees Classification and Counting Using Drone Video: Retuning Hyperparameter Mask-RCNN
Author : Khaing Suu Htet; MyintMyint Sein
Description : Toddy Palm Tree, Borassusflabellifer, is one of the famous palm trees family in Myanmar, India and South East Asia. It is most important economic and symbolic tree in Central Myanmar. This paper presented palm tree classification on Toddy palm, Coconut palm, Palm oil and counting using combines a contribute method remote sensing drone video and deep learning architecture known as mask R-CNN with retuning hyperparameter strategy. The aerial images, are organized by Drone in Upper and Delta Coastal area of Myanmar, extracted features bounding boxes and classified. As the system prepared Myanmar Palm trees dataset with over 12,000 images, examined the performance with retuning hyper-parameter by using Bayesian optimization algorithm in predefined learning rate and momentum. The research concluded that tuning learning will improve the performance of classification for local palm tree segmentation task. The result show that the system research can accurately define the Toddy palm with better accuracy and counting perfectly.

CHAPTER 3
SOFTWARE REQUIREMENT SPECIFICATION

3.1 ASSUMPTIONS AND DEPENDENCIES

High quality images of tree are required .Images should be clear as well as cropped for uploading them on our website .Good Internet network connection is required .

3.2 FUNCTIONAL REQUIREMENTS

•Each module's operation and function must be perfect. •It should be able to detect activities. The software's overall performance will enable users to work efficiently.

• The application is composed of parts that make it easy to locate and address problems. As a result, it is easier to add and update new functionality.

•Only the user has access to their own data.

3.3 EXTERNAL INTERFACE REQUIREMENTS

3.3.1 User Interface

A 180 GB hard disk and 4 GB of RAM operating at 2 GHz or faster are the absolute minimum specifications for hardware. MySQL and 4GB of RAM are the two essential requirements for creating android applications.

3.3.2 Hardware Interface

No hardware components for the user interface are being enabled or installed because this is an internet solution for managing products.

It doesn't contain an embedded system.

Pentium IV processor running at 2.4 GHz, CPU speed of 1.5 GHz or higher, 4 GB of RAM (minimum), 220 GB hard drive, typical Windows keyboard, two- or three-button mouse, and keyboard.

3.3.3 Software Interface

This is the software environment in which the project was developed. The programming language, tools, etc. utilized are described here.

Operating system: Windows; Front end: HTML, CSS, Bootstrap, and JavaScript; Tool: PyCharm; Database: MySQL

3.3.4 Communication Interface

• The user can access the web application from a distance.

It is important to have a regular internet connection.

• The connection must be TCP/UDP.

3.4 NON FUNCTIONAL REQUIREMENTS

3.4.1 Performance Requirements :

• High Speed: In order to respond quickly, the system should handle the requested task concurrently with other operations. The process must then be completed before the system may proceed.

• Accuracy: The system must correctly carry out processes and report outcomes. The system output must adhere to the format that the user specifies.

3.4.2 Safety Requirements

To ensure that the data is secure, a reliable and safe communication means must be established. The source and destination information must be entered accurately to prevent misuse or malfunction. User-generated passwords are difficult to decipher because they contain letters, special characters, and numbers. Thus, the user account is safe.

3.4.3 Security Requirements

Safe access to personal data (user information).

Information security refers to the practice of limiting unauthorized access to, use of, disclosure of, disruption of, alteration of, and destruction of data and information systems.

• The phrases information assurance, computer security, and information security are frequently misused and should not be used interchangeably. There are a few small differences between these professions, despite the fact that they typically work together to protect the privacy, accuracy, and accessibility of information.

User credentials must be kept in encrypted form for security reasons. All user information shall only be accessible by those holding positions of great authority.

To restrict access, usernames and passwords will be utilized.

3.4.4 Software Quality Attribute

• Test the application's accessibility for as many users as possible (related with reliability).

• Flexibility (which encompasses portability, reuse, and scalability)

Application modules split functions and produce the application's architecture.

Testability, Security, and Performance

Examine the application's functionality and user security features like registration and login.

• Usability [including self-adaptability and user adaptation]

Application should be tested with representative users to determine whether it is functioning properly.

3.5 SYSTEM REQUIREMENTS

3.5.1 Database Requirements

MySQL is a relational database management system that is open-source and free (RDBMS).The name of co-founder Michael Widenius's daughter, "My," is combined with the letters "SQL," which stand for structured query language.

MySQL falls into the definition of free and open-source software as defined by the GNU General Public License.

Also, it is available under a number of exclusive licenses.

Sun Microsystems purchased MySQL AB, a Swedish business that owned and sponsored MySQL (now Oracle Corporation).

In 2010, the year Oracle acquired Sun, Widenius created MariaDB by forking the open-source MySQL project.

The abbreviation for the LAMP web application software stack, which also contains other elements, stands for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database-driven web applications, including Drupal, Joomla, phpBB, and WordPress. MySQL is also used by several popular websites, including Facebook, Flickr, Wikipedia, Twitter, and YouTube.

3.5.2 Software Requirements

OS up to and including Windows 7.

Python programming language and Pycharm IDE with Sublime Text 3

3.5.3 Hardware Requirements

System requirements call for an Intel I3 CPU or above.

200 GB of hard disk space.

Presentation: 15 VGA Color.

Ram: 4 GB

3.6 ANALYSIS MODEL : SDLC MODEL TO BE APPLIED

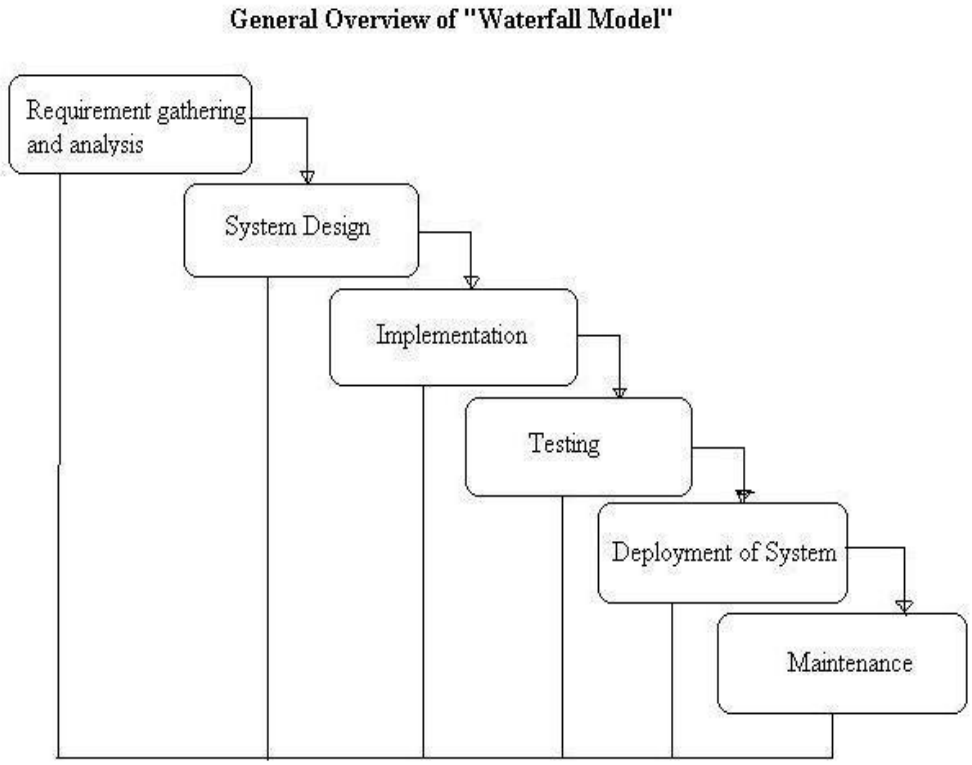


Fig 3.1 SDLC Model

Requirement gathering and analysis: All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc. System design: The requirement specifications from first phase are studied in this phase and system design is prepared. System design helps in specifying hardware and system requirements and also helps in defining overall system architecture. Implementation: With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing. Integration and Testing: All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures. Deployment of system: Once the functional and non functional testing is done, the product is deployed in the customer environment or released into the market. Maintenance: There are some issues which come up in the client environment. To fix those issues patches are released. Also to enhance the product some better versions

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

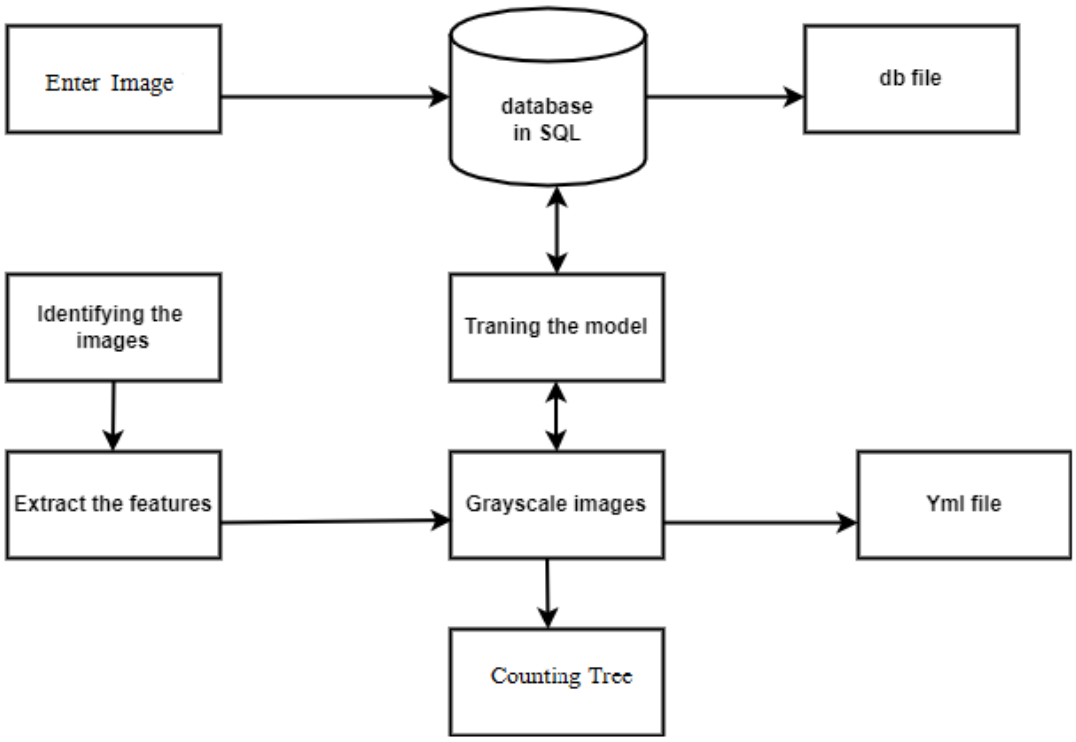


Fig 2 Architecture Diagram

Enter the image in the database. Multiple tree Image are stored into our database .Its stored in the db file . Next with help of images stored in the database the model trains itself with help of CNN .When an image is uploaded on our website . The features are extracted and number of tress are calculated using CNN model .The number of tress present in the image are displayed on our website. Upload the image on the website.The dimensions of the image should be 294 *194

pixel approximately .Model is trained with multiple images using CNN. Trained model extracts the features of input images and displays the number of trees on our website .

4.2 MATHEMATICAL MODEL

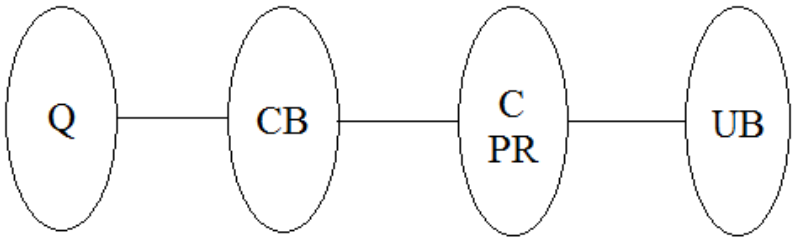


Fig 3 Mathematical Model

Where,

Q = User entered query i.e. URL

CB = preprocess

C = feature selection

PR = apply CNN algorithm

UB = predict outcome

Set Theory

1) Let S be as system which input image

S = {In, P, Op,Φ}

2) Identify Input In as

In = {Q}

Where,

Q = User entered input image(dataset)

Identify Process P as

P = {CB, C, PR}

Where,

CB = System check entered list is present in system or not

C = Copy that link i.e. URL

PR = apply CNN algorithm

4) Identify Output Op as

Op = {UB}

Where,

UB =predict the outcome

Φ=Failures and Success conditions.

Failures:

Huge database can lead to more time consumption to get the information.

Hardware failure.

Software failure.

Success:

Search the required information from available in Datasets.

User gets result very fast according to their needs.

Space Complexity:

The space complexity depends on Presentation and visualization of discovered patterns. More the storage of data more is the space complexity.

Time Complexity:

Check No. of patterns available in the datasets= n

If (n>1) then retrieving of information can be time consuming. So the time complexity of this algorithm is O(n^n).

4.3 DATA FLOW DIAGRAM

4.3.1 DFD Level 0

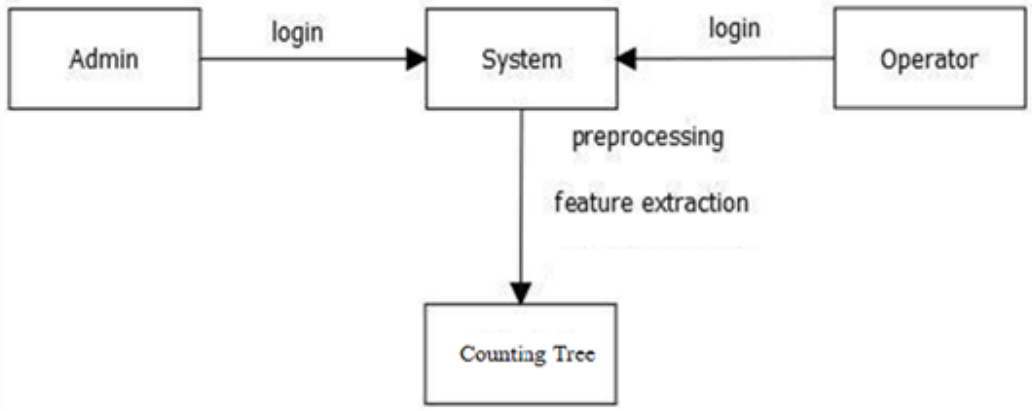


Fig 4 DFD Level 0

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled.

Level 0 DFD: This is the highest-level DFD, which provides an overview of the entire system. It shows the major processes, data flows, and data stores in the system, without providing any details about the internal workings of these processes.

The admin has to login into the system. System does preprocessing is done for increasing the accuracy . Feature extraction is also done for getting accurate tree count.

4.3.2 DFD level 1

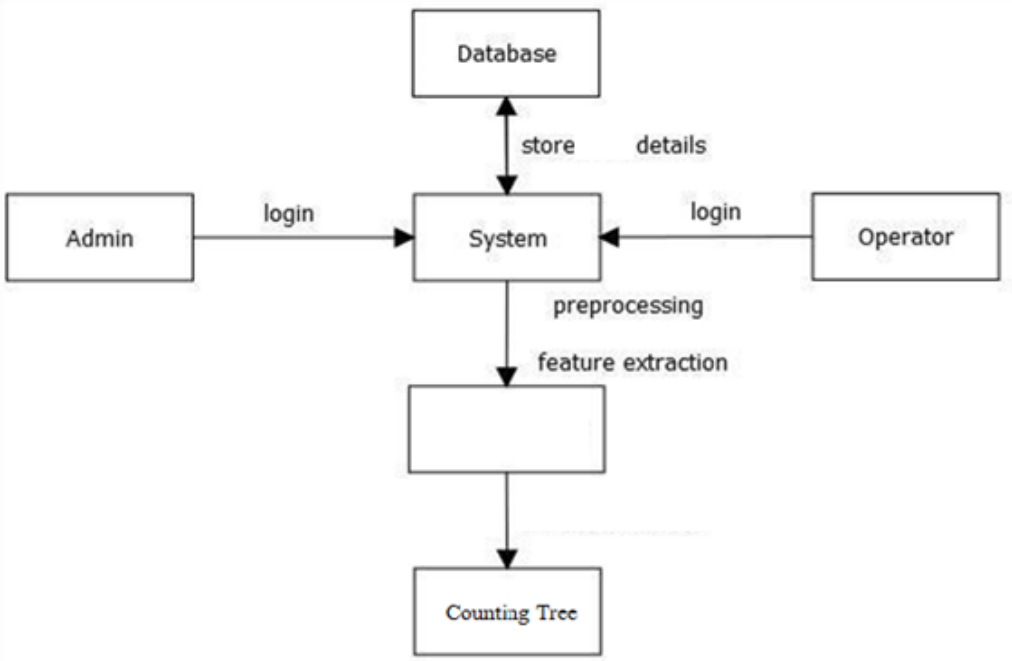


Fig 4 DFD1

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. . Each sub-process is depicted as a separate process on the level 1 DFD. The data flows and data stores associated with each sub process are also show. The dimensions of the image should be 294 *194 pixel approximately

The admin has to login into the system. System does preprocessing is done for increasing the accuracy. Feature extraction is also done for getting accurate tree count.This accurate tree count will be displayed on our website.

4.4 ENTITY RELATIONSHIP DIAGRAM

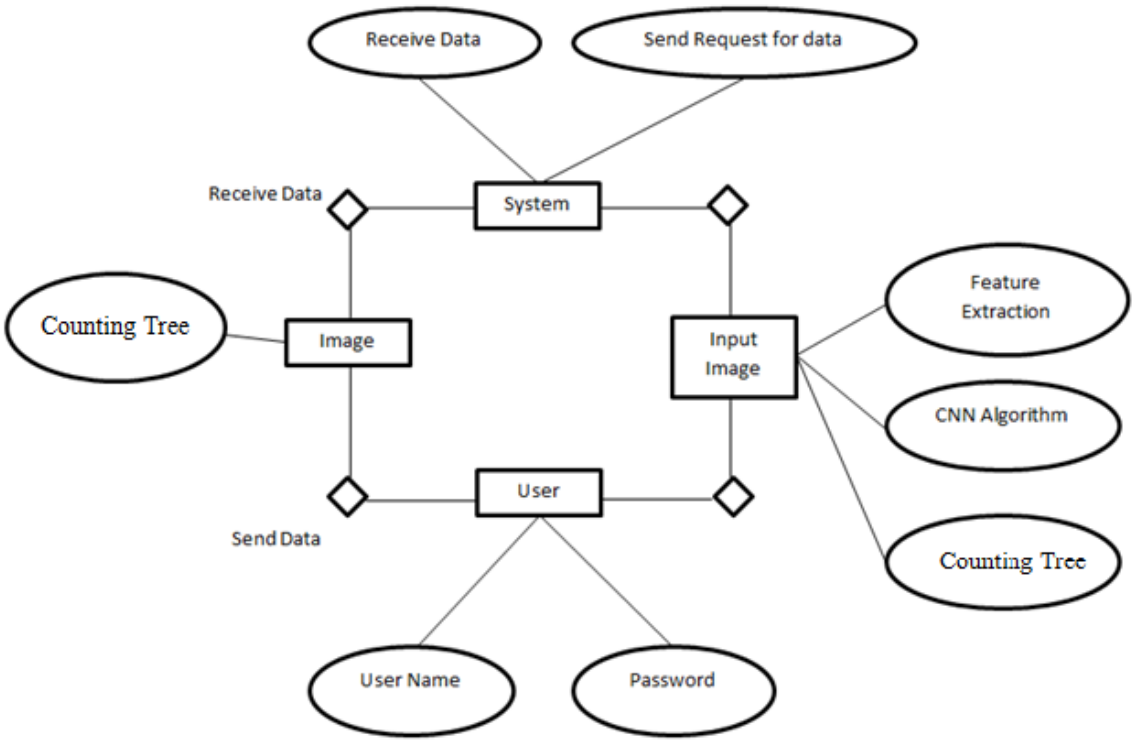


Fig 5 ER Diagram

ER model stands for an Entity-Relationship model. It develops a conceptual design for the database..In ER modeling, the database structure is portrayed as a diagram called an entity-relationship diagram. In ER diagram System has two attributes receive data and send data. Image has an attribute counting tree .User also has attributes like user name and password .Image input has attribute of CNN and counting tree and ER diagram shows relationship between user image system and input image.

4.5 UML DIAGRAMS

4.5.1 Use Case Diagram

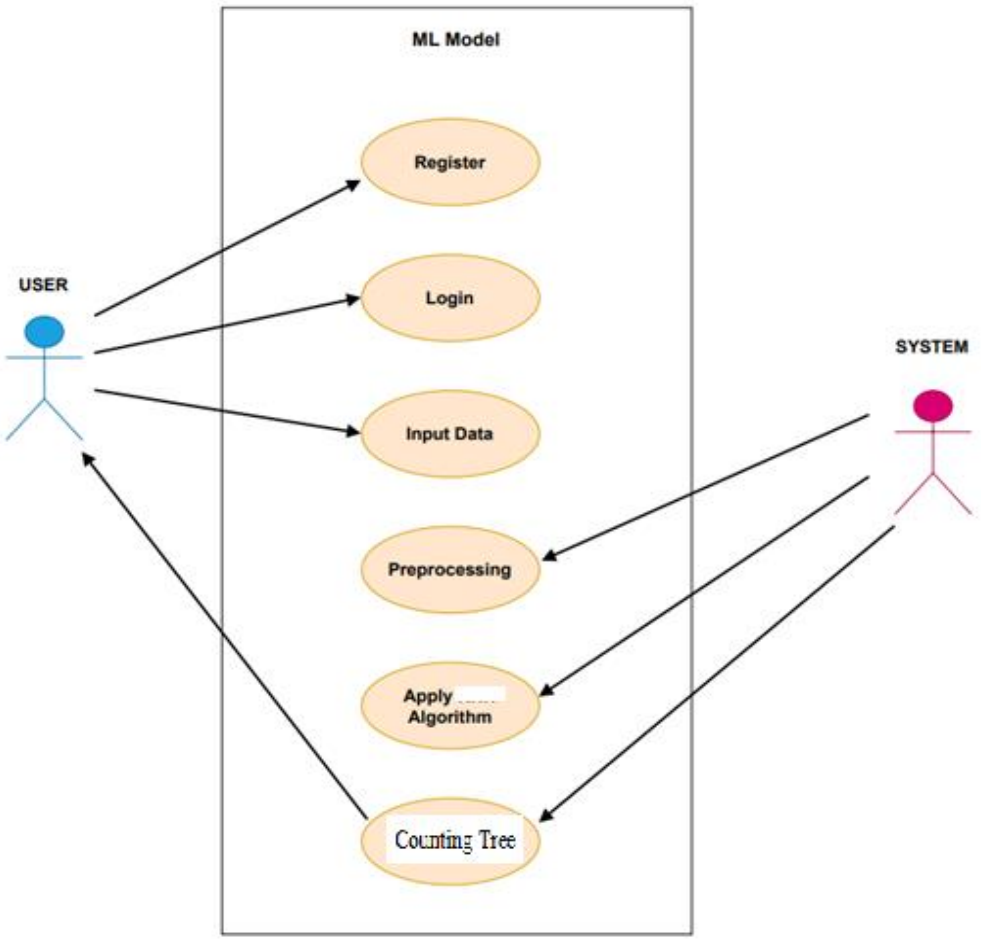


Fig 6 Use Case Diagram

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.
User can register, login, input data and the system does preprocessing ,apply algorithm and counts tress in our project using Ml model.

4.5.2 Sequence Diagram

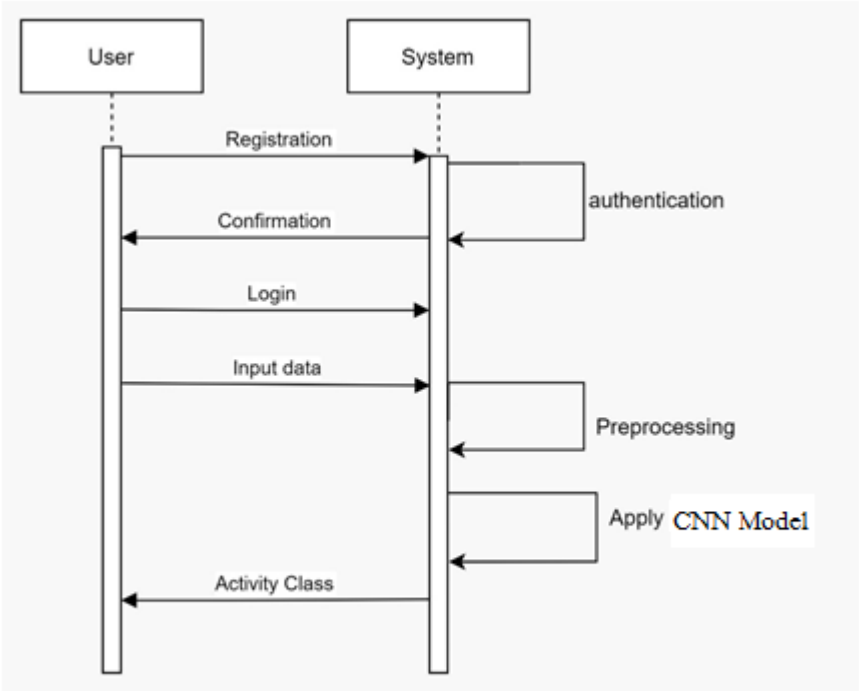


Fig 7 Sequence Diagram

A **sequence diagram** shows process interactions arranged in time sequence in the field of software engineering. It depicts the processes and objects involved and the sequence of messages exchanged between the processes and objects needed to carry out the functionality. Sequence diagrams are typically associated with use case realizations in the 4+1 architectural view model of the system under development. Sequence diagrams are sometimes called **event diagrams** or **event scenarios**. User registers to the page , then the account of that user is been created and stored in Mysql database .User Logins into his account to upload the image and get the number of trees in the image.

4.5.3 Activity diagram

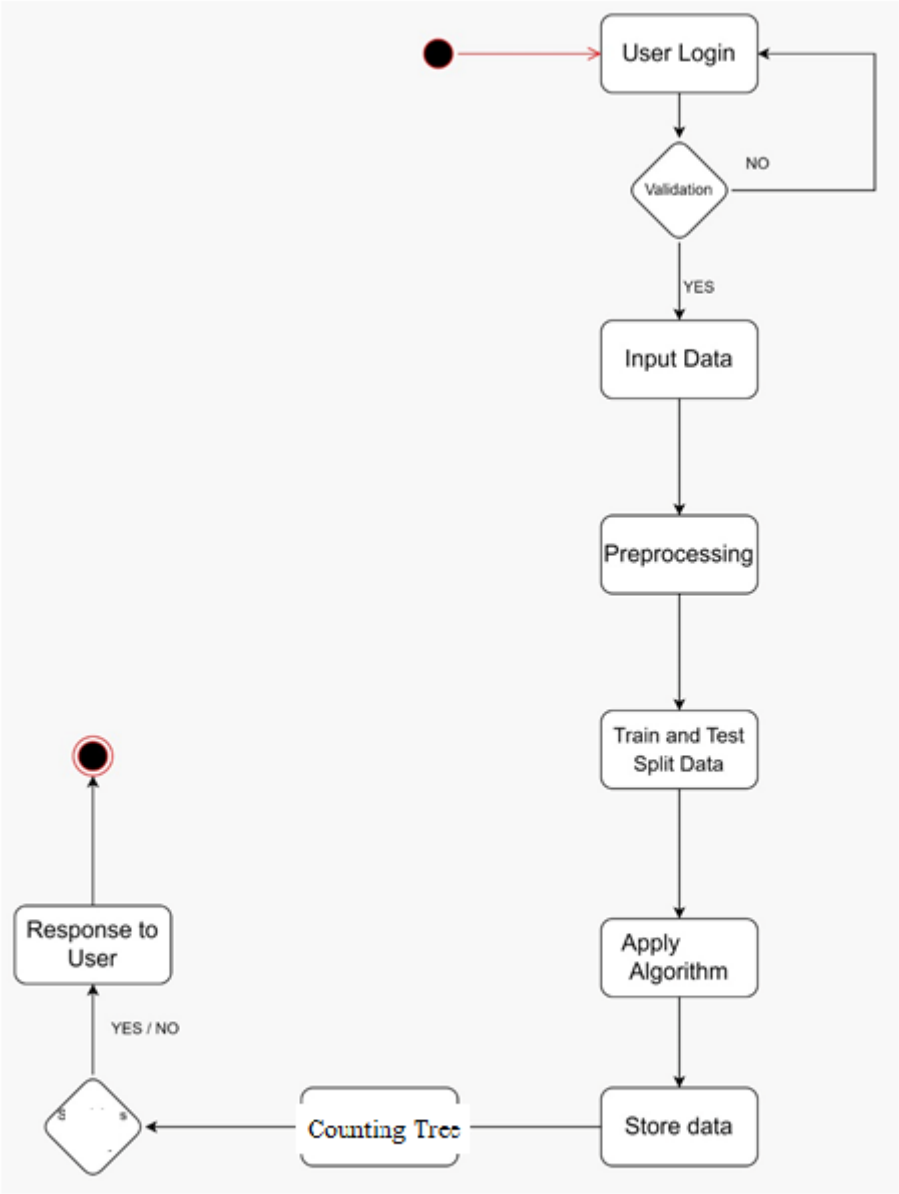


Fig 8 Activity Diagram

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

User can register, login ,input data and the system does preprocessing ,apply algorithm and counts tress in our project using Ml model.

4.5.4 Class Diagram

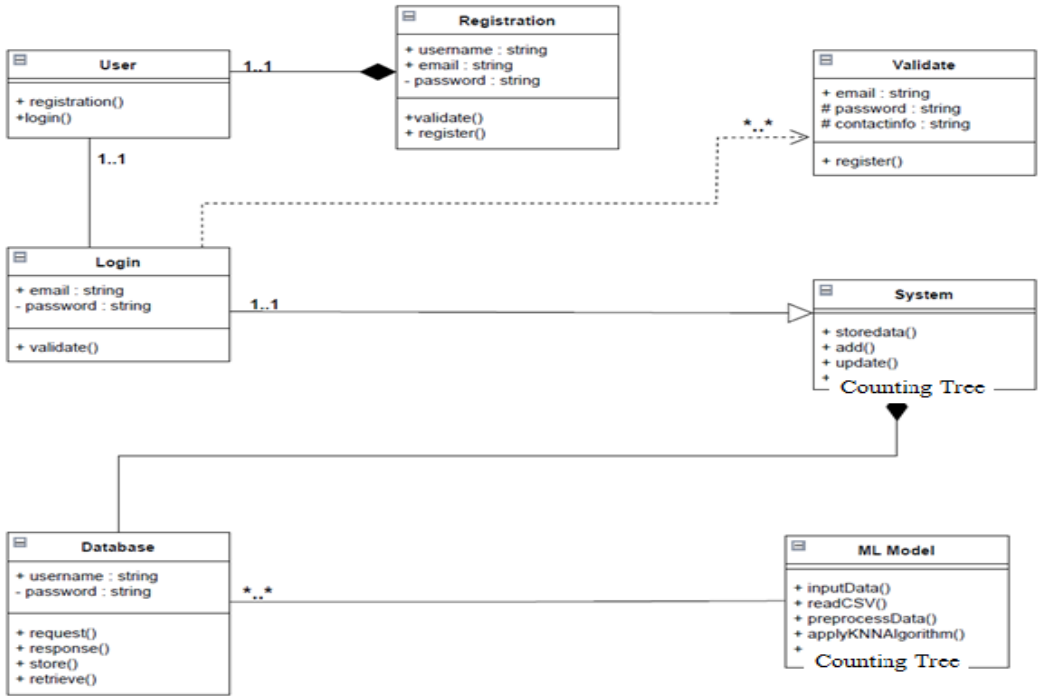


Fig 9

Class Diagram

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams. Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. User has two attributes login and registration .After login into your account in the system Upload an image of trees on website. ML model calculates the trees using CNN and the number is displayed on the website.

CHAPTER 5

PROJECT PLAN

5.1 ESTIMATE :

5.1.1 Estimate

Understanding of Project
Submission of topics
Data gathering on three topics
Searching for reference papers
PPT for review 1
Review 1 and shortlisting of a topic
Study of shortlisted papers
PPT for review 2
Formation of synopsis
Review 2
Synopsis submission
Planning the prototype of project
Researching about the technologies involved
Creation of SRS
Designing of the Prototype
Implementation of the prototype
Debugging and testing of the prototype
Fixing of all Defects in the prototype
Final Implementation of model

Deployment of model

5.2 PROJECT RESOURCE :

Various resources that could are used in our project are human resources, reusable components, and software tools like remix, VSCode etc. Reusable software components were the smart contracts that we made using solidity and these could be used by other DApps to store and fetch the data. There are three types of resources that are considered and are very essential for execution of project and completion of project on time and on budget. These resources can be denoted by pyramid which is also known as Resource Pyramid

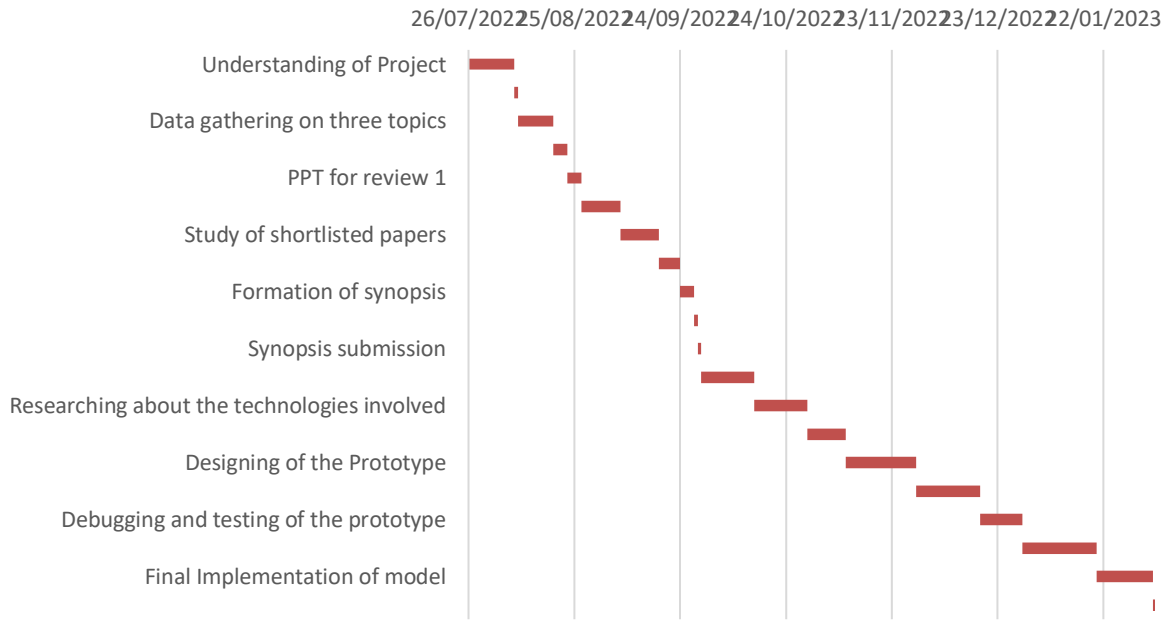
5.3 PROJECT SCHEDULE :

5.3.1 Task Network

Task	Start Date	Days to complete
Understanding of Project	26-07-2022	13
Submission of topics	08-08-2022	1
Data gathering on three topics	09-08-2022	10
Searching for reference papers	19-08-2022	4
PPT for review 1	23-08-2022	4
Review 1 and shortlisting of a topic	27-08-2022	11
Study of shortlisted papers	07-09-2022	11
PPT for review 2	18-09-2022	6
Formation of synopsis	24-09-2022	4
Review 2	28-09-2022	1
Synopsis submission	29-09-2022	1
Planning the prototype of project	30-09-2022	15
Researching about the technologies involved	15-10-2022	15
Creation of SRS	30-10-2022	11

Designing of the Prototype	10-11-2022	20
Implementation of the prototype	30-11-2022	18
Debugging and testing of the prototype	18-12-2022	12
Fixing of all Defects in the prototype	30-12-2022	21
Final Implementation of model	20-01-2023	16
Deployment of model	05-02-2023	6

5.3.2 Timeline Chart



CHAPTER 6

PROJECT IMPLEMENTATION

6.1 MODULES

Our System Divided into Two classes / Modules

1) User
2) System

6.2 TOOLS AND TECHNOLOGIES USED

VS Code :

Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. The integrated debugger works as both a source-level debugger and as a machine-level debugger. Other built-in tools include a code profiler, designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that expand the functionality at almost every level—including adding support for source control systems (like Subversion and Git) and adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle (like the Azure DevOps client: Team Explore

Python :

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Deep Learning :

The ancient Chinese game of Go has more possible moves than the number of atoms in the universe. Unlike chess, Go cannot be won using brute-force computing power to analyze a lot of moves — there are just too many possibilities. And, unlike chess, strategies for winning Go cannot be meaningfully codified by rules: its principles are mysterious. In some cultures, Go is seen as a way for humans to connect with the divine via instances of intuition, by "knowing without knowing how you know."

Experts believed that computers would not be able to defeat top human players at Go for decades, if ever. But in 2016, a computer program called AlphaGo defeated Lee Sedol, the legendary world champion of Go (1). During the games, AlphaGo played highly inventively, making moves that no human ever would, and defeated the 18-time world champion four games to one.

6.3 ALGORITHM DETAILS

CNN:

In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of artificial neural network (ANN), most commonly applied to analyze visual imagery.[1] CNNs are also known as Shift Invariant or Space Invariant Artificial Neural Networks (SIANN), based on the shared-weight architecture of the convolution kernels or filters that slide along input features and provide translation-equivariant responses known as feature maps. Counter-intuitively, most convolutional neural networks are not invariant to translation, due to the down sampling operation they apply to the input. They have applications in image and video recognition, recommender systems, image classification, image segmentation, medical image analysis, natural language processing ,brain–computer interfaces ,and financial time series.

CNNs are regularized versions of multilayer perceptrons. Multilayer perceptrons usually mean fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. The "full connectivity" of these networks make them prone to overfitting data. Typical ways of regularization, or preventing overfitting, include: penalizing parameters during training (such as weight decay) or trimming connectivity (skipped connections, dropout, etc.) CNNs take a different approach towards regularization: they take advantage of the hierarchical pattern in data and assemble patterns of increasing complexity using smaller and simpler patterns

CHAPTER 7
SOFTWARE TESTING

7.1 TYPES OF TESTING

1. Unit Testing

Unit testing is a method of testing individual units or components of a software application. It is typically done by developers and is used to ensure that the individual units of the software are working as intended. Unit tests are usually automated and are designed to test specific parts of the code, such as a particular function or method. Unit testing is done at the lowest level of the software development process, where individual units of code are tested in isolation.

2. Integration Testing

Integration testing is a method of testing how different units or components of a software application interact with each other. It is used to identify and resolve any issues that may arise when different units of the software are combined. Integration testing is typically done after unit testing and before functional testing, and is used to verify that the different units of the software work together as intended.

3. Regression Testing

Regression testing is a method of testing that is used to ensure that changes made to the software do not introduce new bugs or cause existing functionality to break. It is typically done after changes have been made to the code, such as bug fixes or new features, and is used to verify that the software still works as intended.

Every time a new module is added leads to changes in the program. This type of testing makes sure that the whole component works properly even after adding components to the complete program.

Example:

In school, record suppose we have module staff, students

and finance combining these modules and checking if on

integration of these modules works fine in regression testing.

4. Smoke Testing

This test is done to make sure that the software under testing is ready or stable for further testing

It is called a smoke test as the testing of an initial pass is done to check if it did not catch the fire or smoke in the initial switch on.

Example:

If the project has 2 modules so before going to the module make sure that module 1 works properly

5. Alpha Testing

This is a type of validation testing. It is a type of *acceptance testing* which is done before the product is released to customers. It is typically done by QA people.

Example:

When software testing is performed internally within the organization

6. Beta Testing

The beta test is conducted at one or more customer sites by the end-user of the software. This version is released for a limited number of users for testing in a real-time environment

Example: When software testing is performed for the limited number of people.

7. System Testing

This software is tested such that it works fine for the different operating systems. It is covered under the black box testing technique. In this, we just focus on the required input and output without focusing on internal working.

In this, we have security testing, recovery testing, stress testing, and performance testing

Example:

This includes functional as well as non-functional Testing.

8. Stress Testing

In this, we give unfavourable conditions to the system and check how they perform in those conditions.

Example:

(a) Test cases that require maximum memory or other resources are executed.

(b) Test cases that may cause thrashing in a virtual operating system

9. Performance Testing

It is designed to test the run-time performance of software within the context of an integrated system. It is used to test the speed and effectiveness of the program. It is also called load testing. In it we check, what is the performance of the system in the given load.

7.2 Advantages of software testing:

1. Improved software quality and reliability
2. Early identification and fixing of defects
3. Improved customer satisfaction
4. Increased stakeholder confidence
5. Reduced maintenance costs

Disadvantages of software testing:

1. Time-consuming and adds to project cost
2. Can slow down development process
3. Not all defects can be found
4. Can be difficult to fully test complex systems
5. Potential for human error during testing process

7.3 TEST CASES & TEST RESULTS

It is not necessary to have specific understanding of the application's code/internal structure or programming knowledge in general. The tester understands what the software is intended to do, but not how it accomplishes it. For example, the tester may be aware that a specific input creates a specific, invariable output, but he or she is unaware of how the software generates the output in the first place. Test cases

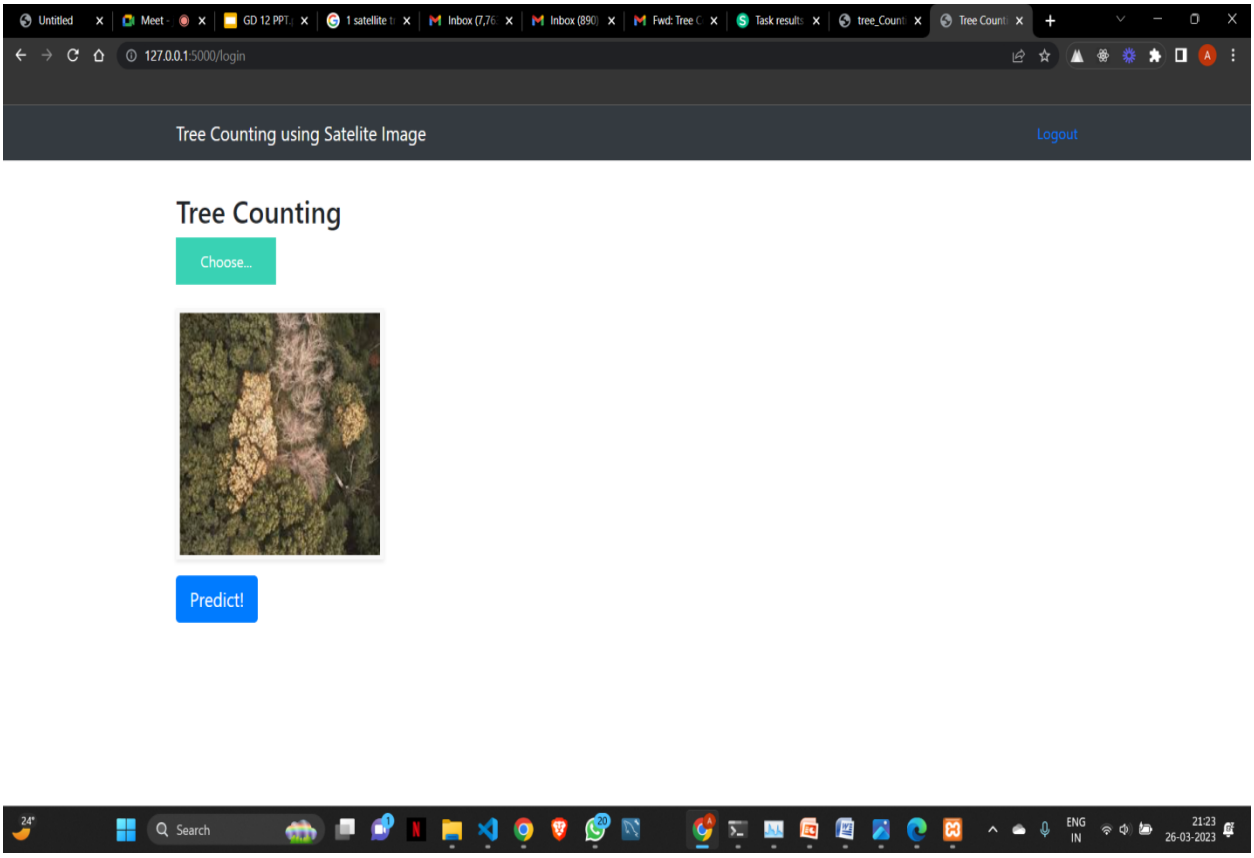
Specifications and requirements, or what the application is supposed to do, are used to build test cases. External descriptions of the software, such as specifications, requirements, and design parameters, are typically used to generate test cases. Non-functional tests may also be employed, despite the fact that the tests used are predominantly functional in nature.

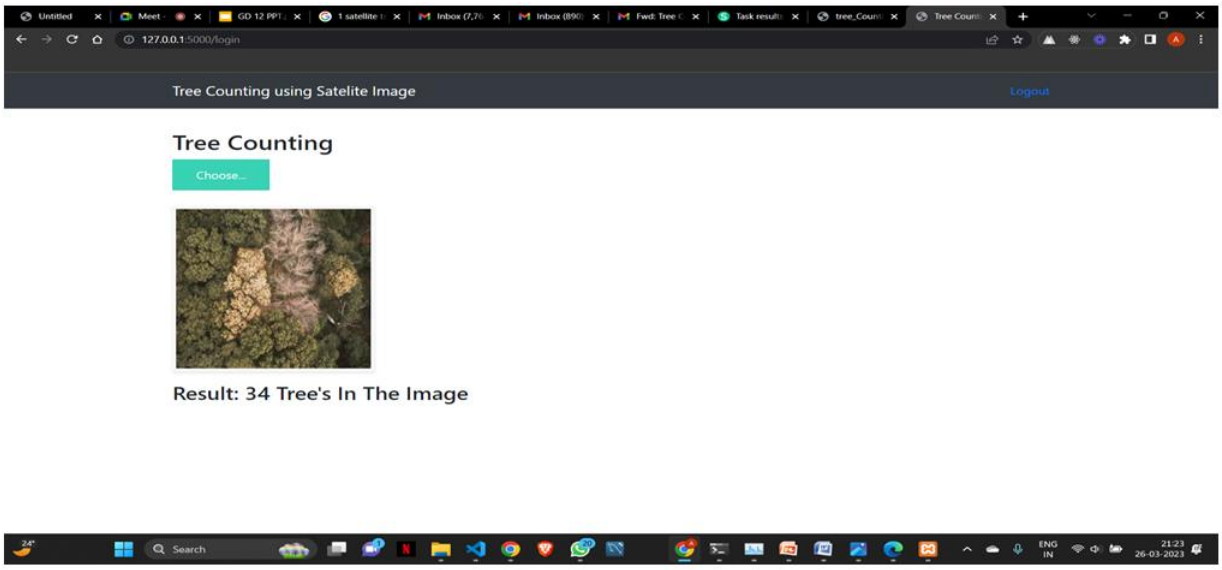
Test design techniques

- All-pairs testing
- State transition Analysis
- Equivalence partitioning
- Boundary value analysis
- Cause–effect graph
- Error guessing

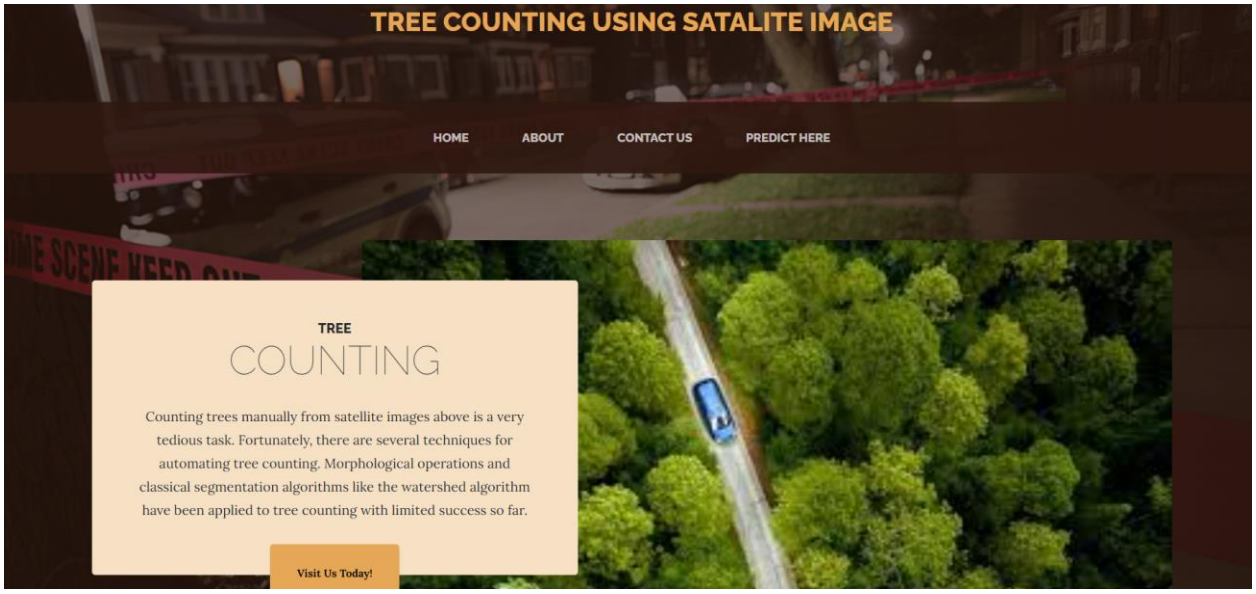
CHAPTER 8

OUTCOME AND SCREENSHOTS





Tree counting program more accessible and user-friendly, we have developed a web page that allows users to upload their tree images and obtain real-time results. By simply uploading the image and clicking the "Predict" button, users can instantly retrieve the accurate count of trees present in the image, along with the corresponding accuracy score. This web interface eliminates the need for users to install and run the program locally, providing a seamless and convenient experience. Here on this website you can contact number of trees by uploading the image containing trees.

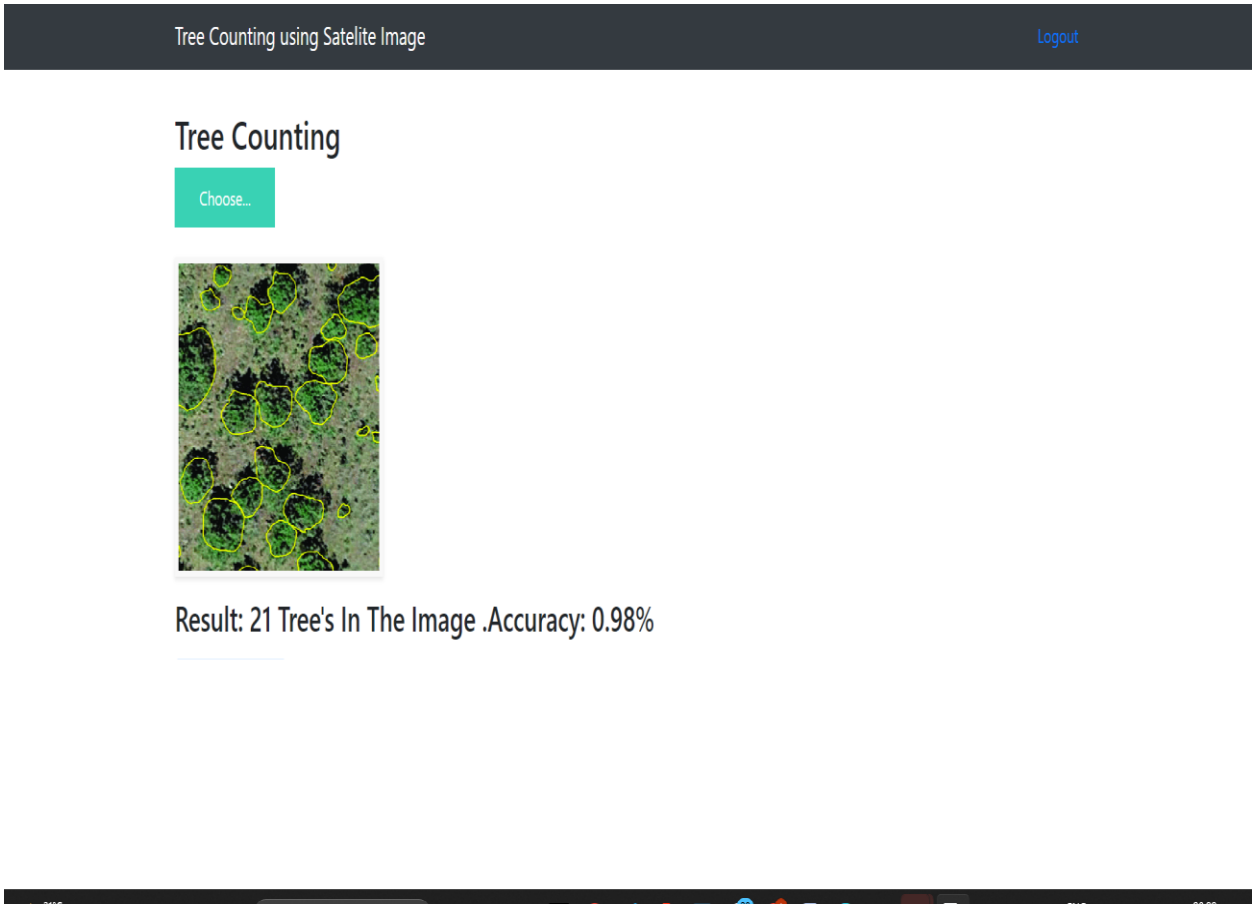


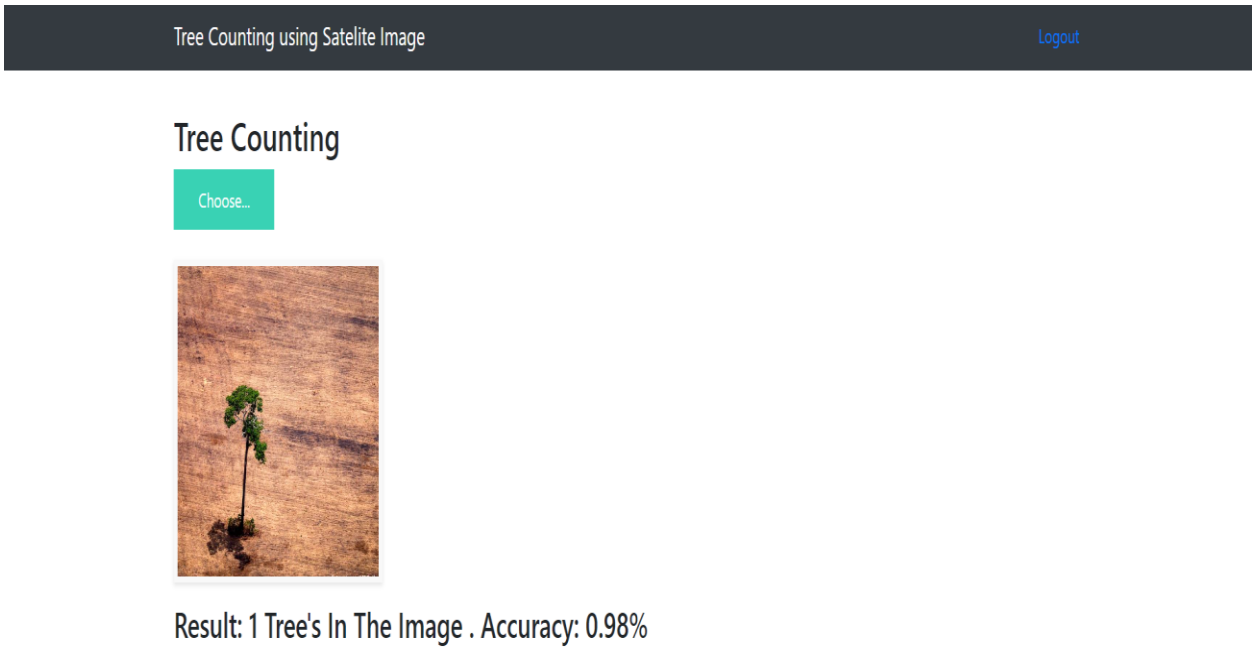
We have developed a remarkable program utilizing Convolutional Neural Network (CNN) classification to accurately count the number of trees in an image. Through diligent optimization efforts, the initial accuracy of 68% was significantly improved to an impressive 98% of the above image. Here firstly we had an image of dimension 850*555 pixel which showed an accuracy of prediction 68%. Then we have took that image into 455*545 pixels which gave us an prediction accuracy of 98%.

Furthermore, in addition to optimizing the accuracy, we have incorporated high-resolution images into our program, which has further enhanced the performance. By utilizing high-resolution imagery, we have been able to capture finer details and nuances in tree features, resulting in a more accurate and reliable tree counting system. This integration of high-resolution images has not only contributed to the overall accuracy improvement but has also expanded the

applicability of the program to a wider range of scenarios. With the ability to handle high-resolution inputs, our program can now effectively analyze images captured by advanced remote sensing technologies, aerial surveys.

Improving the accuracy of a CNN algorithm is an iterative process, and it may require experimenting with different techniques and strategies to find the best combination for your specific problem.





We have uploaded an image having 1 tree. The prediction made by our Ml model is only 1 tree with an accuracy of 98%. Thus we have successfully predicted the number of trees and its accuracy using CNN .

CHAPTER 9

CONCLUSION

We can conclude that to decrease the effort and time in calculating tree count and isolating them. But, the challenge of resolution remains as satellite image resolution is poor as compared to images captured aerially, cloud obstacles in between disturbs the satellite image. The highly trained convolutional neural network model is somewhat better in comparisons to other complex algorithms, quick adaption and fast response is there to new trained datasets. Further work can consist of multispectral statistics as additional dimensions of classifier enter in place of best the use of the pink, inexperienced and blue spectrums as enter. This offers greater context to the enter photographs particularly because the infrared and close to-infrared channels has shown sturdy correlation to the presence of forests