Deforestation Monitoring Tool

Submitted in partial fulfillment of the requirements

of the degree of

Bachelor of Technology in Information Technology

By

Atharva Gitaye (24101A2001)

Anish Bandal (24101A2005)

Mukta Redij (24101A2007)

Under the Guidance of

Dr. Varsha Bhosale

Department of Information Technology

Autonomous Institute affiliated University of Mumbai

Vidyalankar Institute of Technology

Wadala(E), Mumbai-400437

University of Mumbai

2024-25

CERTIFICATE OF APPROVAL

This is to certify that the project entitled

Deforestation Monitoring Tool

is a bonafide work of

Atharva Gitaye (24101A2001)

Anish Bandal (24101A2005)

Mukta Redij (24101A2007)

submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the

degree of **Bachelor of Technology** in **Information Technology.**

(Dr. Varsha Bhosale)
Project Guide

Dr. Vidya Chitre
Head of Department

Dr. Sangeeta Joshi
Principal

PROJECT REPORT APPROVAL FOR S. E.

This project report entitled Deforestation Monitoring Tool by

1. ***Atharva Gitaye (24101A2001)***
2. ***Anish Bandal (24101A2005)***
3. ***Mukta Redij (24101A2007)***

is approved for the degree of ***Bachelor of Technology in Information Technology.***

1.________________________________

Name and Signature External Examiner

2.________________________________

Name and Signature Internal Examiner

Date:

Place: Mumbai

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Name of student	Roll No.	Signature
1. Atharva Gitaye	24101A2001	
2. Anish Bandal	24101A2005	
3. Mukta Redij	24101A2007	

Date:
Place: Mumbai

ACKNOWLEDGEMENT

Before presenting our final year project work entitled "**Deforestation Monitoring Tool**", we would like to convey our sincere thanks to the people who guided us throughout the course for this project work.

First, we would like to express our sincere thanks to our beloved Principal Dr. Sangeeta Joshi for providing various facilities to carry out this project.

We would like to express our immense gratitude towards our Project Guide Dr. Varsha Bhosale for constant encouragement, support, guidance, and mentoring at the ongoing stages of the project and report.

We would like to express our sincere thanks to our H.O.D. Dr. Vidya Chitre, for the encouragement, co-operation, and suggestions for progressing stages of the report.

Finally, we would like to thank all the teaching and non-teaching staff of the college, and our friends, for their moral support rendered during the course of the reported work, and for their direct and indirect involvement in the completion of our report work, which made our endeavour fruitful.

Date:
Place: Mumbai

ABSTRACT

Deforestation, driven by rapid urbanization and land-use changes, is a pressing environmental concern, particularly in semi-urban and metropolitan regions. The accurate and timely monitoring of vegetation loss is critical for urban planning, ecological conservation, and sustainable development. Traditional methods of deforestation tracking are often static, limited in resolution, or inaccessible to non-technical users. To address this gap, this project introduces a two-part Deforestation Monitoring Tool that utilizes Remote Sensing and GIS technologies to detect and visualize vegetation loss using satellite-derived NDVI (Normalized Difference Vegetation Index).

The project is divided into two main components. The first component is a web-based application developed using Flask (Python) as the backend, Google Earth Engine (GEE) for satellite data processing, and Folium for dynamic map visualization. Users can input coordinates, a buffer distance, and a base year to generate an interactive vegetation loss map. The application processes Sentinel-2 imagery, performs NDVI comparisons between the selected year and the current year, and classifies vegetation loss into severity categories, which are rendered in real-time through an interactive HTML map.

The second component focuses on region-specific NDVI analysis for the Mumbai Metropolitan Region, including Mumbai, Navi Mumbai, and Thane. This module utilizes GEE's cloud computing capabilities to calculate annual NDVI composites from 2016 to 2024, detect vegetation loss relative to the reference year (2024), and export results as GeoTIFFs and statistical CSV files. The spatial outputs are then analyzed and visualized using QGIS, a free and open-source desktop GIS platform.

The tool provides both user-specific and regional deforestation insights with high spatial resolution and user-friendly accessibility. It demonstrates a scalable and replicable model for environmental monitoring. Future scope includes frontend enhancements, polygon-based analysis, and support for downloadable reports to improve usability and analytical flexibility.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

Abbreviation	Description
NDVI	Normalized Difference Vegetation Index
GEE	Google Earth Engine
GIS	Geographic Information System
ROI	Region of Interest
AOI	Area of Interest
API	Application Programming Interface
HTML	HyperText Markup Language
QGIS	Quantum Geographic Information System
SR	Surface Reflectance
CSV	Comma Separated Values
MODIS	Moderate Resolution Imaging Spectroradiometer

1. INTRODUCTION

1.1 Problem Definition

Deforestation poses a serious threat to ecological sustainability, contributing to global environmental challenges such as climate change, biodiversity loss, and habitat destruction. With increasing urbanization, especially in rapidly developing metropolitan regions like Mumbai, Navi Mumbai, and Thane, natural vegetation is often cleared to make way for infrastructure and industrial expansion. As a result, monitoring and managing vegetation loss have become critical for environmental planning and conservation.

Although satellite data is widely available, stakeholders such as researchers, planners, and policymakers often lack the tools to analyse this data effectively in a localized, timely, and user-specific manner [3],[4]. Traditional manual mapping approaches are time-consuming, limited in scope, and may not be updated frequently. Furthermore, the complexity of remote sensing data and GIS tools can make them inaccessible to non-expert users [4].

There is a clear need for a solution that simplifies the process of analysing vegetation loss, provides both localized and regional insights, and leverages advanced technologies for accurate and efficient detection.

1.2 Aim and Objective

The primary aim of this project is to design and implement a **Deforestation Monitoring Tool** that utilizes **Remote Sensing** and **Geographic Information System (GIS)** technologies to analyse and visualize vegetation loss using satellite-derived **NDVI (Normalized Difference Vegetation Index)** data.

This tool is divided into two integrated modules:

1. Web-Based Interactive Application

- Allows users to input geographical coordinates, a search radius (in kilometres), and a base year.

- Generates a real-time interactive map that visualizes vegetation loss from the selected base year to the current year.
- Uses Google Earth Engine (GEE) to process Sentinel-2 satellite data, compute NDVI, and classify vegetation loss severity.
- Renders the output using Folium with custom legends, overlays, and control elements.

2. Region-Specific Deforestation Analysis (Mumbai, Navi Mumbai, Thane)

- Focuses on a fixed urban area (Mumbai Metropolitan Region).
- Performs annual NDVI-based analysis from the year 2016 to 2024.
- Generates classified vegetation loss maps using GEE and exports the results as GeoTIFF files.
- Visualizes results in QGIS for detailed interpretation, supported by statistical reports for each year-to-year comparison.

The key objectives are as follows:

- To integrate GEE for remote sensing data access and preprocessing.
- To calculate NDVI and detect vegetation loss through time-series analysis.
- To classify vegetation loss severity using NDVI thresholds.
- To deliver dynamic visualizations via a web-based application.
- To generate downloadable geospatial outputs and statistical summaries for regional insights.

1.3 Organization of the Report

This report is organized into the following chapters:

- **Chapter 1: Introduction**
  Provides background context, outlines the project's problem statement, and presents the overall aim, objectives, and structure of the report.

- **Chapter 2: Review or Literature**

  Discusses prior work in deforestation monitoring using remote sensing and GIS, highlighting methods and tools relevant to the project.
- **Chapter 3: Requirement Specification**

  Details the hardware and software requirements, feasibility study, and estimated cost of development.
- **Chapter 4: Project analysis and design**

  Describes the architecture of both the web-based and QGIS-based systems, including component interactions and data flow.
- **Chapter 5: Methodology**

  Explains the step-by-step workflow for NDVI computation, cloud masking, buffer-based region extraction, and deforestation severity classification.
- **Chapter 6: Implementation details**

  Provides the code-level explanation of the web backend (Flask and GEE), frontend visualization (Folium), and QGIS outputs.
- **Chapter 7: Result Analysis**

  Presents maps, visuals, and comparative data for user-generated areas and the Mumbai-Thane-Navi Mumbai regional study.
- **Chapter 8: Conclusion and Future Scope**

  Summarizes findings, discusses limitations, and suggests potential enhancements such as frontend optimization and ML-based classification.
- **References and Appendices**

  Include citations, reports, certificates, GitHub repository link, and supporting documentation.

2. REVIEW OF LITERATURE

2.1 Literature Survey

Tracking deforestation through remote sensing and geospatial analysis has been a critical area of research for decades, given its implications for climate change, biodiversity conservation, and sustainable urban development [2],[4]. Over the years, several platforms, tools, and scientific methods have been developed to monitor forest loss, assess vegetation health, and analyze land cover changes.

This chapter reviews the key technologies and systems that have historically supported deforestation monitoring, identifies their limitations, and explains how the present work advances this domain. It also highlights how the integration of Google Earth Engine (GEE), NDVI analytics, and interactive GIS visualization addresses the gap between raw data availability and actionable, user-driven insights.

2.2 Existing Systems

a. Global Satellite Monitoring Platforms

Global systems such as **NASA's Fire Information for Resource Management System (FIRMS)** and **Global Forest Watch (GFW)** provide near real-time forest fire alerts and forest loss tracking using MODIS and Landsat satellite data [2],[10],[11]. These platforms have made large-scale forest monitoring accessible to the public.

Limitations:

- Lower spatial resolution (250m to 30m) is insufficient for detecting fine-scale vegetation loss in semi-urban or urban areas.
- Largely static and generalized, limiting localized, high-resolution assessments.
- Minimal options for user-driven analysis or region-specific customization.

b. NDVI-Based Research Applications

NDVI has long been a standard metric for assessing vegetation health [1]. Several academic studies have employed NDVI to track seasonal and long-term changes in forest cover using satellite data such as MODIS, Landsat, and Sentinel [1],[12].

Limitations:

- Tools are typically research-oriented and require technical expertise to operate.
- NDVI values must be interpreted manually or through complex geospatial software.
- Often lack a visual, user-friendly interface or web-based accessibility.

c. Google Earth Engine (GEE) Applications

Google Earth Engine enables cloud-based processing of satellite imagery and has empowered many scientific projects related to deforestation, climate analysis, and land use classification [3]. GEE supports large-scale spatial computations using scripting environments.

Limitations:

- Primarily code-based; users must write JavaScript or Python scripts to utilize GEE functionalities.
- Lacks built-in user interfaces or web deployment by default.
- Non-technical users find it difficult to access or interact with GEE-powered applications without a frontend.

d. Web GIS and Mapping Tools

Web mapping libraries such as **Leaflet.js**, **Folium**, and **OpenLayers** are frequently used for geospatial data visualization in browser environments [9]. They allow interactive maps to be rendered from geospatial data sources.

Limitations:

- Commonly used for static or preprocessed geospatial data.
- Limited real-time processing capabilities and backend integration.
- Typically not tailored for vegetation-specific monitoring or analysis.

2.3 Current Implementation and Improvements

The **Deforestation Monitoring Tool** developed in this project addresses the limitations of existing systems by integrating **real-time satellite data analysis with user-centric visualization tools**. It combines the capabilities of **Google Earth Engine**, **Flask**, and **Folium** to provide an interactive and accessible platform for both general users and domain experts.

The system consists of two distinct modules:

1. Web-Based Interactive Vegetation Loss Tool

- Allows users to input latitude, longitude, search radius (in kilometers), and a base year.
- Performs NDVI-based analysis using Sentinel-2 imagery with 10-meter spatial resolution.
- Classifies vegetation loss severity based on NDVI difference between the base and current year.
- Visualizes the result on an interactive Folium map with legends, buffer overlays, and controls.
- Designed to be accessible for non-expert users such as educators, urban planners, and students.

2. QGIS-Based Regional Deforestation Analysis (Mumbai Metropolitan Region)

- Focuses on pre-defined urban areas (Mumbai, Navi Mumbai, and Thane).
- Uses GEE to generate yearly NDVI composites from 2016 to 2024.
- Identifies vegetation loss trends by comparing each year's NDVI with that of a reference year (2024).
- Outputs include red-tile highlighted deforested zones, true color composites, GeoTIFF exports, and CSV-based statistical summaries.
- Maps are visualized and analyzed in QGIS for detailed, high-resolution interpretation and documentation.

Key Improvements Introduced:

Feature	Improvement
User Interactivity	Enables input of location, radius, and year for custom analysis.
High Spatial Resolution	Uses Sentinel-2 data (10m resolution) for detailed vegetation detection.
Dynamic Visualization	Generates NDVI maps in real-time with interactive controls and legends.
Localized Focus	Goes beyond global coverage by offering focused analysis for Mumbai Metropolitan Region.
Accessibility	Provides a non-scripted interface for broader accessibility beyond the research community.

Table 2.1. Key Improvements

This implementation bridges the gap between research-grade tools and practical, real-world usability. It not only enhances the accuracy and granularity of deforestation detection but also democratizes access to satellite-based environmental monitoring for a wider audience.

3. REQUIREMENT SPECIFICATION

3.1 Introduction

The Deforestation Monitoring Tool is a dual-component system designed to analyze, detect, and visualize deforestation trends using satellite imagery and geospatial data. It leverages Remote Sensing and GIS techniques to assess vegetation changes across time and space. The two core modules of this system include:

1. **Web-Based Vegetation Loss Detection Tool** – An interactive application allowing users to input geographic coordinates, search radius, and base year to generate a vegetation loss map using NDVI comparisons, powered by Google Earth Engine (GEE) and rendered with Folium on a dynamic Flask backend.
2. **QGIS-Based Regional Analysis for Mumbai Metropolitan Region** – A comprehensive analysis module that uses Sentinel-2 and Landsat-8 satellite imagery processed via GEE scripts to compute NDVI over multiple years (2016–2024), generate comparative GeoTIFF layers, and visualize them using QGIS for interpretation and reporting.

This chapter outlines the hardware and software requirements, feasibility analysis, and cost estimation necessary for development, deployment, and usage of the system.

3.2 Hardware Requirements

Client-Side (End User)

- **Processor**: Intel Core i3 or higher
- **RAM**: Minimum 4 GB
- **Storage**: Minimum 500 MB free space
- **Internet Connection**: Required for accessing the web-based interface and maps

Server-Side (Development & Hosting)

- **Processor**: Intel Core i5 or higher

- **RAM**: 8 GB or more
- **Storage**: At least 100 GB (to support GIS files and image exports)
- **Operating System**: Ubuntu 20.04 LTS / Windows Server
- **GPU (Optional)**: Recommended for accelerated raster data processing and rendering

---

3.3 Software Requirements

Client-Side

- **Web Browser**: Latest versions of Google Chrome, Mozilla Firefox, or Microsoft Edge
- **No additional installations required** (web app is browser-based)

Server-Side / Developer Environment

- **Operating System**: Ubuntu 20.04 LTS or compatible Linux distribution
- **Programming Language**: Python 3.8 or above
- **Framework**: Flask (for RESTful API and backend routing)
- **GIS Tools**:
  - **QGIS** (for visualization of GeoTIFFs and spatial data) [7].
  - **Google Earth Engine (GEE)** (for satellite data access and NDVI computation) [3].
- **Spatial Data Libraries**:
  - **Earth Engine Python API** – For accessing and processing satellite imagery and NDVI calculations directly from Google Earth Engine.
  - **Folium** – For generating interactive maps with overlays, legends, and controls using Leaflet.js in Python [9].
  - **Requests / JSON / Logging** – For managing API requests, responses, and logging events in the Flask backend.

    - **Flask-CORS** – For enabling Cross-Origin Resource Sharing to allow frontend and backend communication [8].

- **Map Rendering**:
    - **Folium (Python)** – For interactive leaflet-based map generation
- **Web Server**: Apache or Nginx (optional, for deployment)

Database (Optional for reports/metadata storage)

- **SQLite** – For storing statistical reports or user queries

---

3.4 Feasibility Study

Technical Feasibility

The project is technically feasible due to its reliance on mature, well-supported open-source technologies such as Flask, Google Earth Engine, and QGIS. The integration between the backend (Python) and GEE for processing NDVI, along with Folium for rendering dynamic maps, has been successfully implemented and tested.

Operational Feasibility

The system is designed for ease of use by non-technical users such as environmental researchers, educators, and planners. The web interface requires minimal input and displays results in a comprehensible, visual manner. The QGIS maps are packaged as downloadable files and can be interpreted using standard GIS tools.

Economic Feasibility

By utilizing open-source platforms (GEE, QGIS, Python), the overall cost of development is significantly minimized. Potential expenses are limited to hosting, optional cloud storage, and domain registration.

Legal and Environmental Feasibility

The project does not rely on proprietary or restricted data. All satellite imagery is obtained from publicly available repositories (e.g., Sentinel-2 via GEE). The tool supports environmental objectives, promoting sustainable land management and forest conservation.

3.5 Cost Estimation

Component	**Estimated Cost (INR)**
Server Hosting (1 year)	₹8,000 – ₹15,000
Domain Name Registration (1 year)	₹800 – ₹1,200
Cloud Storage / Backup (Optional)	₹4,000 – ₹8,000
Miscellaneous (Setup, Power, etc.)	₹2,500 – ₹4,000
Total Estimated Cost	₹15,300 – ₹28,200

Table 3.1. Cost Estimation for Hosting and Architecture

Note: All development tools and libraries (QGIS, Flask, GEE, Python libraries) are open-source and free to use. Hosting costs may vary depending on cloud provider (e.g., AWS, GCP, DigitalOcean).

4. PROJECT ANALYSIS AND DESIGN

4.1 Use Case Diagram

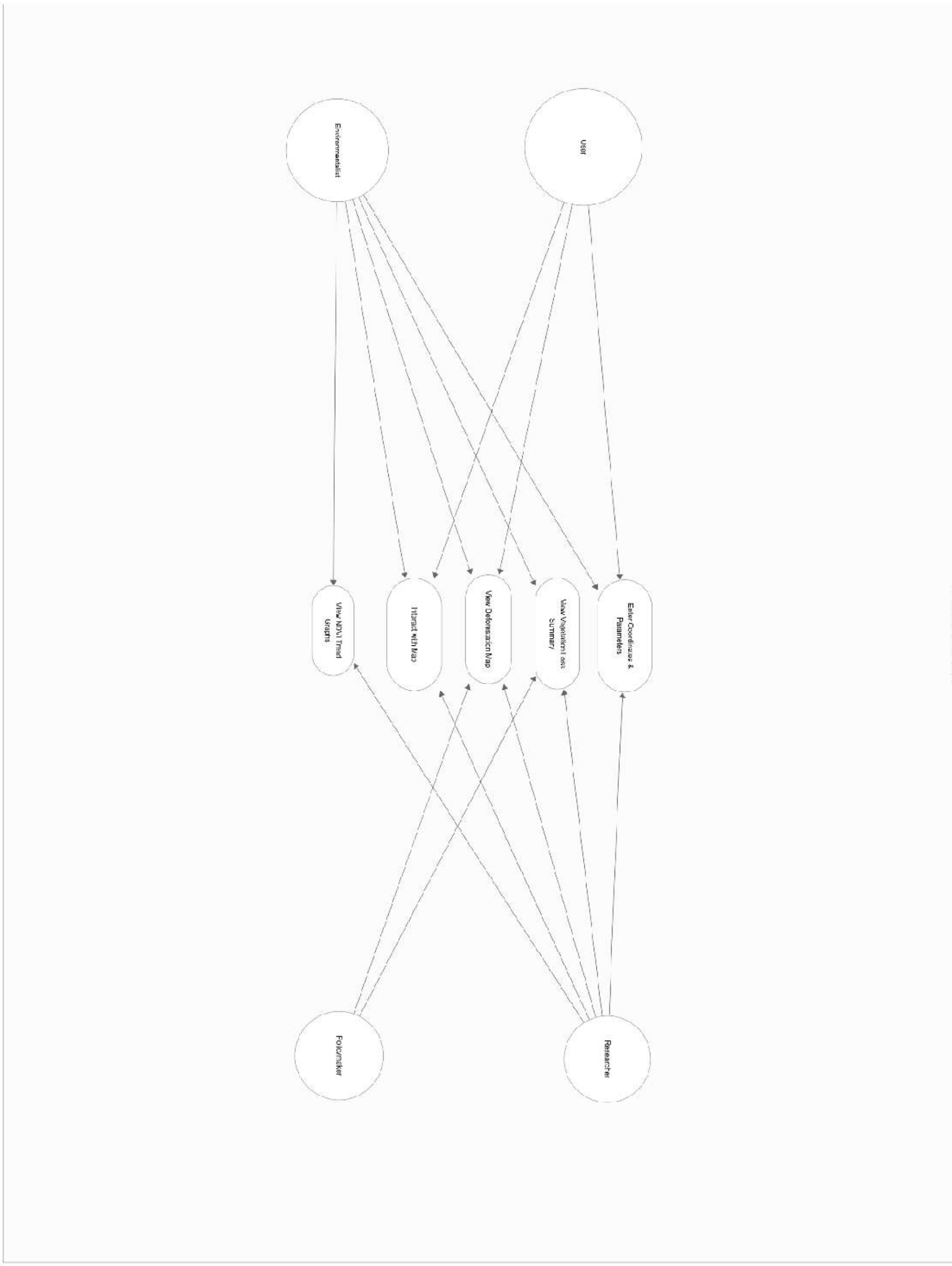


Figure 4.1 Use Case Diagram of the Deforestation Monitoring Tool

4.2 System Architecture Design

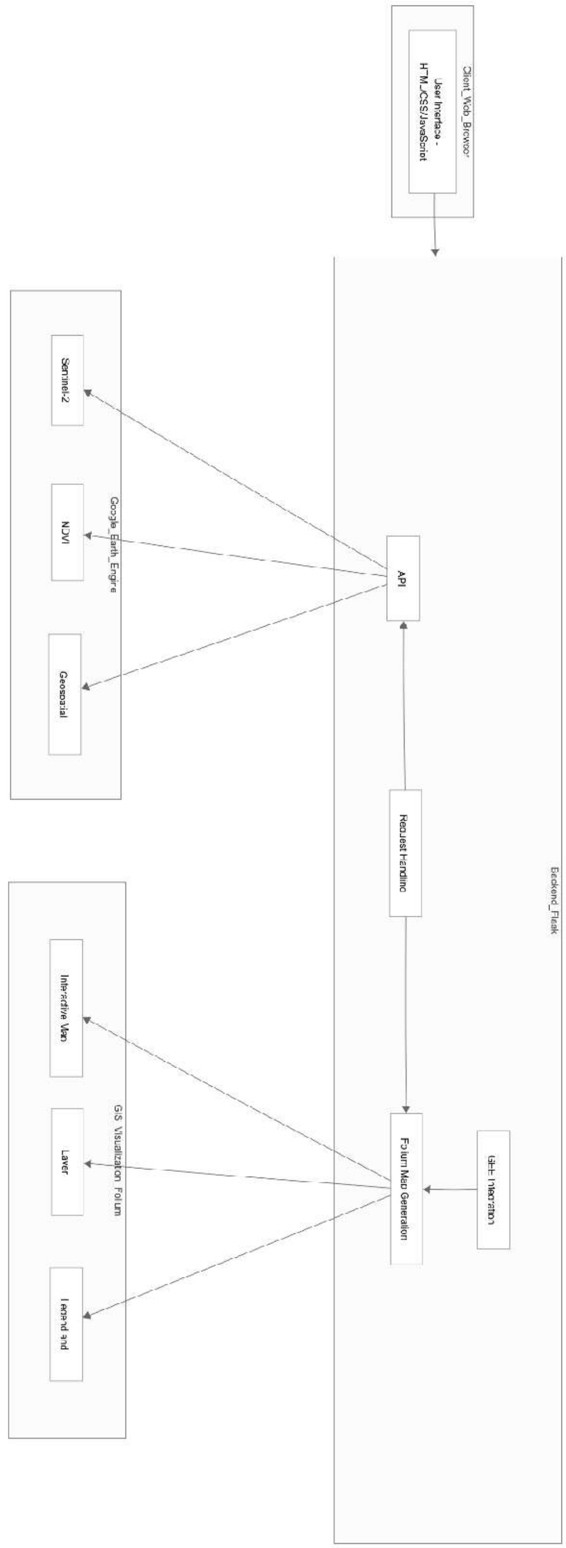


Figure 4.2 System Architecture Diagram

4.3 Flowchart

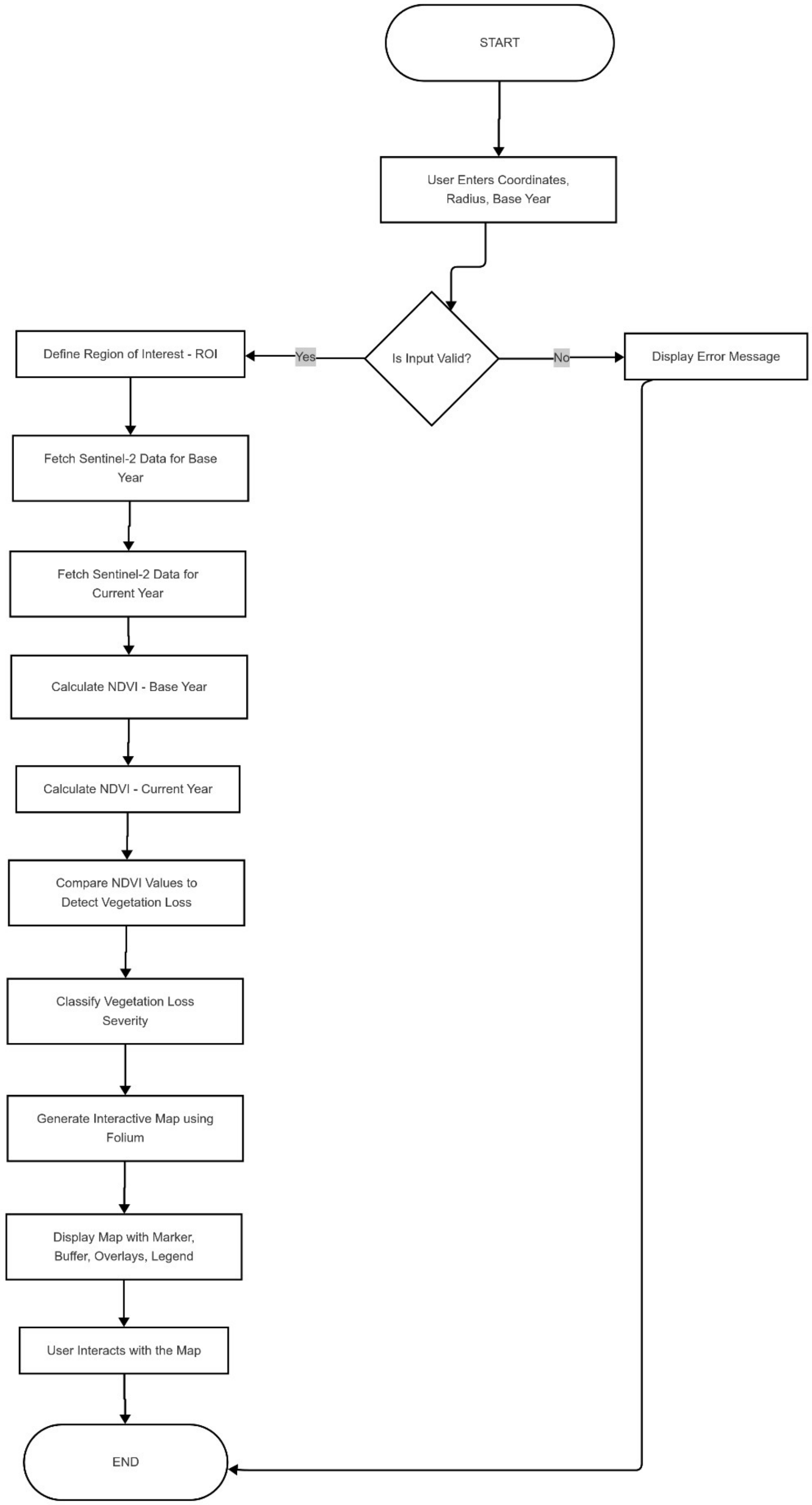


Figure 4.3 Flowchart Representing NDVI Calculation and Map Generation Logic

5. METHODOLOGY

5.1 Overview

The methodology adopted for the Deforestation Monitoring Tool is based on a combination of cloud-based remote sensing analysis and interactive geospatial visualization. The project is divided into two main components:

1. **Remote Sensing and GIS-Based Time-Series Analysis** (focused on the Mumbai Metropolitan Region using Google Earth Engine and visualized in QGIS).
2. **Interactive Web Application Development** (using Flask, Google Earth Engine API, and Folium for user-defined NDVI-based vegetation loss detection).

Each component follows a structured workflow involving satellite data processing, vegetation change detection using NDVI, spatial visualization, and user interaction.

5.2 Regional GIS-Based Time-Series Analysis (Mumbai Region)

This component involves the generation of NDVI-based vegetation loss maps over multiple years using **Google Earth Engine (GEE)** and visualization using **QGIS**.

Step 1: Defining the Study Area

- The Area of Interest (AOI) was defined as a polygon covering **Mumbai, Navi Mumbai, and Thane**.
- The geometry was manually constructed within the GEE Code Editor using latitude-longitude boundary coordinates.

Step 2: Satellite Data Acquisition

- Multitemporal imagery was accessed using:
    - **Sentinel-2 SR-Harmonized Collection** for high-resolution NDVI analysis (10m) [5].
    - **Landsat-8** for historical NDVI comparison. [6], [12]
- Imagery was filtered based on date, cloud coverage (<60%), and spatial bounds (AOI).

Step 3: Preprocessing and NDVI Calculation

- **Cloud masking** was applied using the QA60 band to remove pixels obscured by clouds or cirrus.
- **NDVI** was calculated using the formula: *(NIR-RED)/(NIR+RED)*
  where NIR = Band 8 and RED = Band 4 for Sentinel-2 [1].
- **Annual NDVI composites** were created using median pixel values for each year (2016 to 2024).

Step 4: Change Detection and Classification

- NDVI from each year was compared with the **reference year 2024** to detect vegetation loss.
- NDVI loss thresholds were applied:
  - Loss > 0.1 → Very Low
  - Loss > 0.2 → Low
  - Loss > 0.3 → Moderate
  - Loss > 0.4 → High
  - Loss > 0.5 → Severe
- Classified deforestation maps were exported as **GeoTIFFs** for QGIS visualization.

Step 5: Visualization and Reporting in QGIS

- Exported NDVI loss layers were imported into **QGIS** for manual analysis.
- Red-tiled areas indicated regions of deforestation.
- Year-wise comparison maps, area summaries, and vegetation loss heatmaps were generated.
- A CSV file was exported containing statistics such as area lost per year (in sq.km and hectares).

5.3 Web-Based Deforestation Detection Tool (Flask + GEE + Folium)

This module allows users to perform dynamic, NDVI-based deforestation analysis for any location by entering a geographic point, radius, and base year.

Step 1: User Input via Web Interface

- The user provides:
    - Latitude and Longitude of the location.
    - Radius in kilometers (study area buffer).
    - A base year for NDVI comparison (2016–2023).

Step 2: Backend API Processing (Flask)

- A **Flask-based REST API** validates user inputs.
- A circular **Region of Interest (ROI)** is generated from the coordinates and radius.
- The backend calls Google Earth Engine via its Python API to:
    - Fetch satellite imagery for both base year and current year.
    - Apply cloud masking and calculate NDVI for both years.
    - Subtract NDVI to compute vegetation loss.
    - Classify severity using defined NDVI thresholds.

Step 3: Vegetation Loss Classification

- Vegetation loss is classified into five categories using NDVI difference values:
    - 1: Very Low (≥ 0.1)
    - 2: Low (≥ 0.2)
    - 3: Moderate (≥ 0.3)
    - 4: High (≥ 0.4)

    - 5: Severe (≥ 0.5)
- Only regions that had vegetation in the base year are considered.

Step 4: Map Rendering with Folium

- A **Folium map** is generated containing:
    - The classified severity layer using GEE Map Tiles.
    - A circle showing the study area radius.
    - A legend indicating severity levels.
    - Interactive map controls (zoom, layers, fullscreen, mouse position).
- The output is rendered as a complete HTML map in the browser.

Step 5: Output Delivery

- Users are able to:
    - View vegetation loss interactively on the map.
    - Observe severity color codes.
    - Interpret results without any technical background.
- (Optional extension) CSV/GeoTIFF downloads may be provided in future versions.

5.4 Summary of Methodology

Component	**Tools & Technologies**	**Purpose**
Satellite Imagery	Google Earth Engine, Sentinel-2, Landsat-8	Vegetation health analysis (NDVI)
Spatial Analysis	NDVI calculation, NDVI difference, classification	Detect and quantify deforestation
Web Backend	Flask, Earth Engine Python API	API handling and NDVI processing
Frontend Map	Folium (Python)	Rendering interactive vegetation loss maps
Regional Visualization	QGIS	Interpretation of classified GeoTIFFs
Output Formats	HTML (Web), GeoTIFF (QGIS), CSV (Statistics)	Delivering analysis in visual and numeric form

Table 5.1. Summary of Methodology

6. IMPLEMENTATION DETAILS

6.1 Overview

This chapter describes the technical implementation of the two main components of the project:

1. **Interactive Web Application** for real-time, user-defined vegetation loss analysis using Flask, Google Earth Engine (GEE), and Folium.
2. **Time-Series NDVI Analysis for the Mumbai Metropolitan Region**, using Google Earth Engine scripting and QGIS for post-processing and visualization.

The implementation details include backend and frontend workflows, algorithmic logic for NDVI processing, and outputs in the form of maps and statistics.

6.2 Module 1 – Web Application (Flask + GEE + Folium)

6.2.1 Technologies Used

- **Programming Language**: Python 3.8+
- **Framework**: Flask (RESTful backend)
- **Geospatial Platform**: Google Earth Engine (via Earth Engine Python API)
- **Mapping Library**: Folium (based on Leaflet.js)
- **Libraries**: ee, folium, flask, flask_cors, datetime, json, logging

6.2.2 Functional Workflow

1. **Input Interface**:
    - Users provide Latitude, Longitude, Radius (in km), and Base Year.
    - Input validation checks are performed for range and format.
2. **Backend Processing**:
    - A circular Region of Interest (ROI) is created using the provided location and buffer distance.

- GEE fetches Sentinel-2 imagery for both the base year and the current year.
- Cloud masking is applied using the QA60 band.
- NDVI is computed as:

  NDVI = (B8 – B4)/(B8 + B4) or (NIR – RED)/(NIR + RED)
- NDVI images from the two years are compared to calculate NDVI loss.
- Vegetation loss is classified into five severity levels based on NDVI difference thresholds.

3. **Map Generation**:
   - A Folium map is dynamically generated and rendered with:
     - NDVI loss severity layer from GEE.
     - Circle indicating ROI.
     - Center marker with popup details.
     - Legend showing severity classification.
     - Fullscreen, mouse position, and minimap controls.
4. **Output**:
   - Refer Figure 6.1 for output

6.2.3 Sample Code Snippet (Backend - NDVI Logic)

```
def generate_vegetation_loss_map(latitude, longitude, distance, base_year):
    """Generate vegetation loss map using NDVI difference between years."""
    try:
        # Define Region of Interest (ROI)
        distance_meters = distance * 1000
        roi = ee.Geometry.Point([longitude, latitude]).buffer(distance_meters)
        current_year = datetime.now().year
        base_year_composite = get_year_composite(base_year, roi)
```

```
    current_year_composite = get_year_composite(current_year, roi)
    if base_year_composite is None or current_year_composite is None:
        raise ValueError("Missing satellite data for selected years")
    # Calculate NDVI Difference
    ndvi_diff = current_year_composite.select('NDVI').subtract(
        base_year_composite.select('NDVI')).rename('NDVI_diff')
    # Mask vegetation from base year
    vegetation_mask = base_year_composite.select('NDVI').gt(0.2)
    vegetation_loss = ndvi_diff.lt(-0.1).rename('vegetation_loss')
    masked_loss = vegetation_loss.updateMask(vegetation_mask)
    # Classify severity of vegetation loss
    severity = ee.Image(1)
    severity = severity.where(ndvi_diff.lt(-0.2), 2)
    severity = severity.where(ndvi_diff.lt(-0.3), 3)
    severity = severity.where(ndvi_diff.lt(-0.4), 4)
    severity = severity.where(ndvi_diff.lt(-0.5), 5)
    severity = severity.updateMask(masked_loss).rename('severity')
    return severity, roi
except Exception as e:
    logger.error(f"Error generating vegetation loss map: {str(e)}")
    raise
```

6.3 Module 2 – Regional NDVI Analysis (GEE + QGIS)

6.3.1 Area of Interest

- The analysis covers **Mumbai, Navi Mumbai, and Thane**, defined as a polygon geometry in Google Earth Engine.

6.3.2 Time-Series NDVI Workflow

1. **Data Acquisition**:

- Sentinel-2 Surface Reflectance data (2016–2024).
- Landsat-8 used for older years if required.

2. **NDVI Composites**:
   - Median NDVI is calculated annually.
   - Cloud masking ensures data quality.
   - Reference year is fixed as 2024.
3. **Change Detection**:
   - Each year is compared to 2024.
   - NDVI loss calculated as:

     $\Delta NDVI = NDVI_{2024} - NDVI_{Year}$
   - Pixels with NDVI decrease beyond 0.1 are marked as deforested.
   - Forest mask is applied to restrict analysis to vegetated areas.
4. **Export Outputs**:
   - **GeoTIFFs** for deforestation severity layers per year [7].
   - **True color composites** for ground verification [5].
   - **CSV** files with year-wise statistics (area lost in hectares and sq.km).

6.3.3 Visualization in QGIS

- GeoTIFFs are imported into QGIS.
- Deforested areas are highlighted in red.
- NDVI change maps are analyzed spatially.
- CSV data used for creating year-wise bar charts and summaries.

6.4 Sample Output Maps (Screenshots from Web and QGIS Modules)

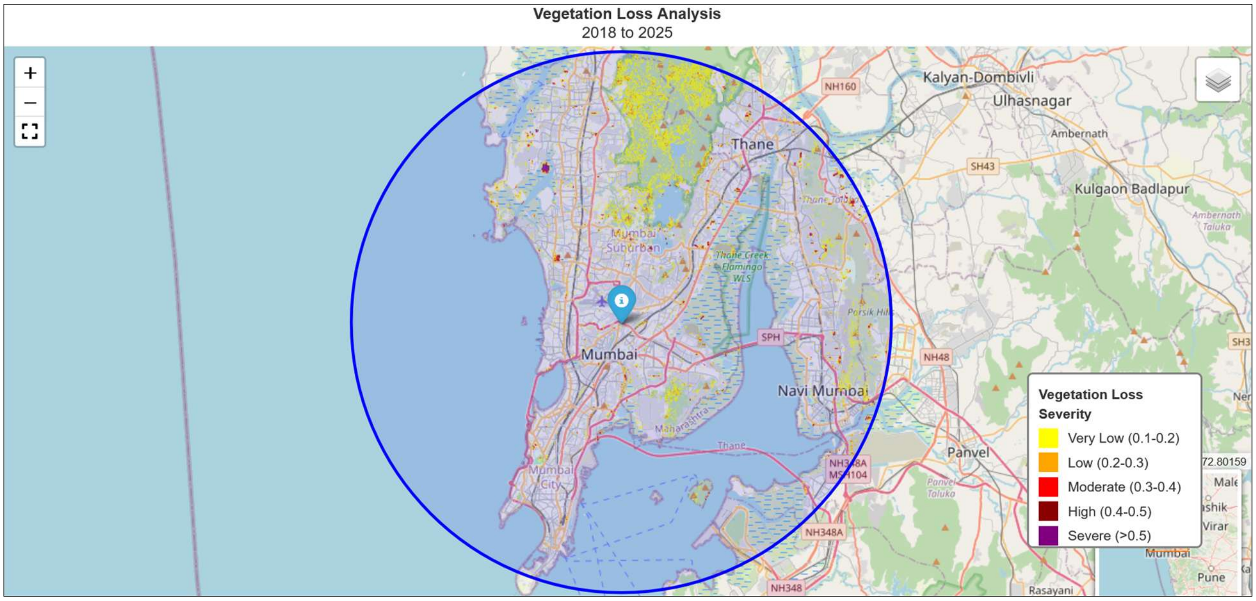


Figure 6.1 Screenshot of Interactive NDVI Loss Map Generated by Web app

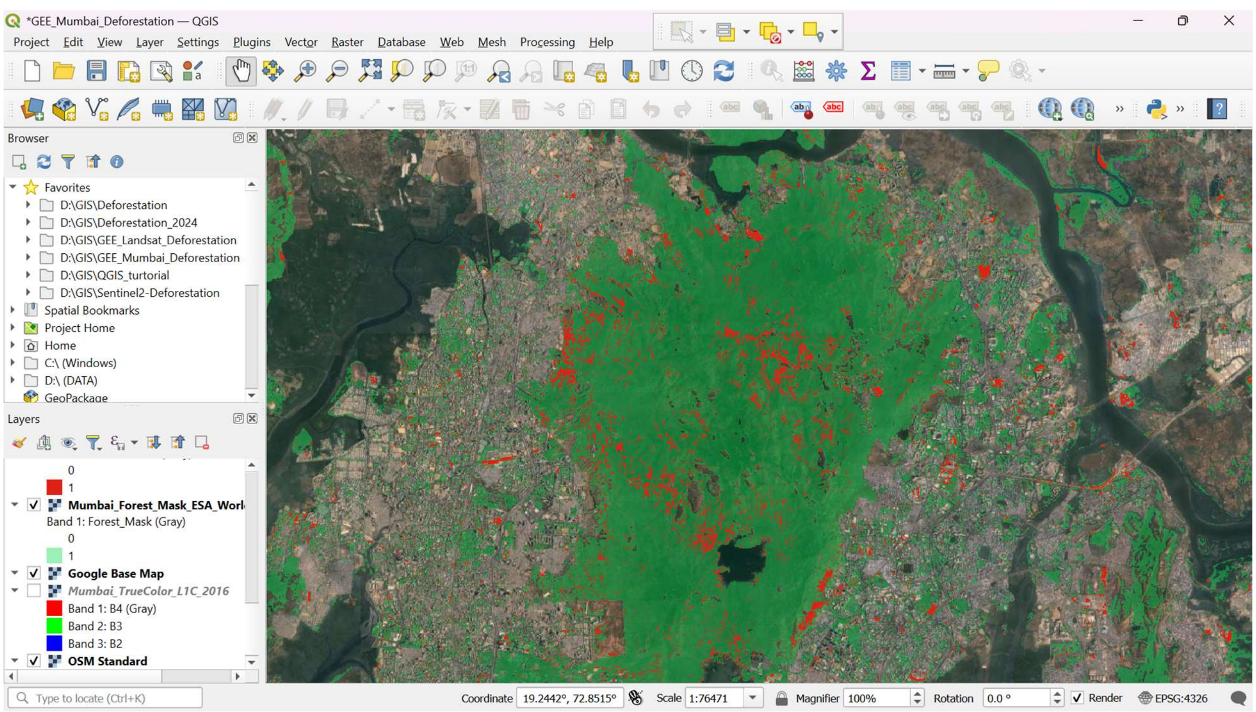


Figure 6.2 Deforestation Map using Sentinel 2 (QGIS).

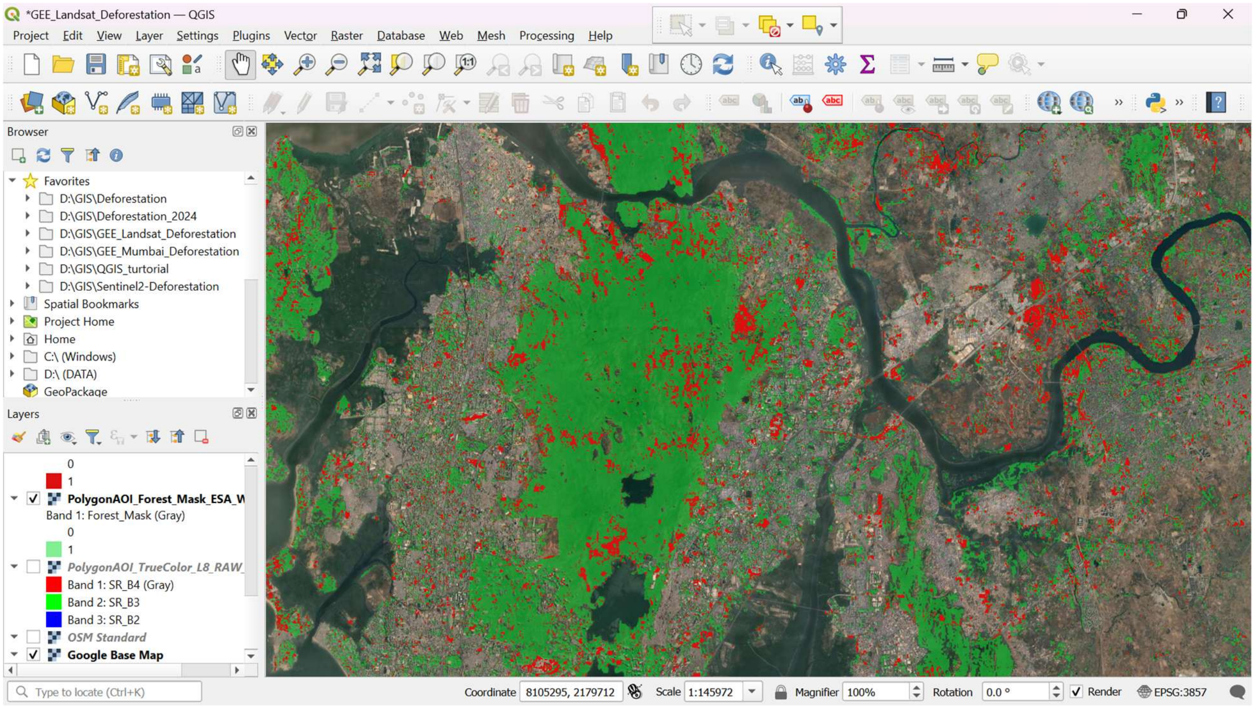


Figure 6.3 Deforestation Map using Landsat 8 (QGIS).

6.5 Challenges Faced and Resolutions

Challenge	**Resolution**
High memory usage in GEE for NDVI export	Used tile scaling and clipped image boundaries
Sentinel-2 cloud interference	Applied QA60 cloud masking for cleaner NDVI results
UI rendering performance (large buffers)	Limited max buffer distance and used tile loading optimizations
Map layer alignment in Folium	Matched EPSG projections and controlled zoom/center params

Table 6.1. Challenges Faced and Resolution

7. RESULT ANALYSIS

7.1 Overview

This chapter presents the output generated from both modules of the Deforestation Monitoring Tool. The results include visual interpretations of vegetation loss using NDVI change detection, deforestation severity classification, and region-wise statistical summaries. Outputs were produced using Google Earth Engine and rendered through a web interface (Folium) and QGIS for spatial analysis.

The results are discussed in two categories:

1. Web-Based Real-Time NDVI Loss Visualization
2. Multi-Year Regional Deforestation Analysis (Mumbai Metropolitan Region)

7.2 Web-Based NDVI Loss Detection Results

This module allows users to dynamically input a geographic location, buffer distance, and a base year. Based on these inputs, vegetation loss from the selected base year to the current year is analyzed and visualized.

7.2.1 Output Description

- The output is rendered as an interactive HTML map.
- The map displays:
    - A circle denoting the region of interest (ROI).
    - A marker at the center point.
    - A raster overlay showing vegetation loss severity (color-coded).
    - A legend for interpreting NDVI loss classes:
        - Yellow: Very Low Loss (0.1–0.2)
        - Orange: Low Loss (0.2–0.3)

- Red: Moderate Loss (0.3–0.4)
- Dark Red: High Loss (0.4–0.5)
- Purple: Severe Loss (>0.5)

7.2.2 Case Example

- **Input Parameters**

  Latitude: 19.2183

  Longitude 72.9781

  Radius: 20

  Base Year: 2018

- **Output**:

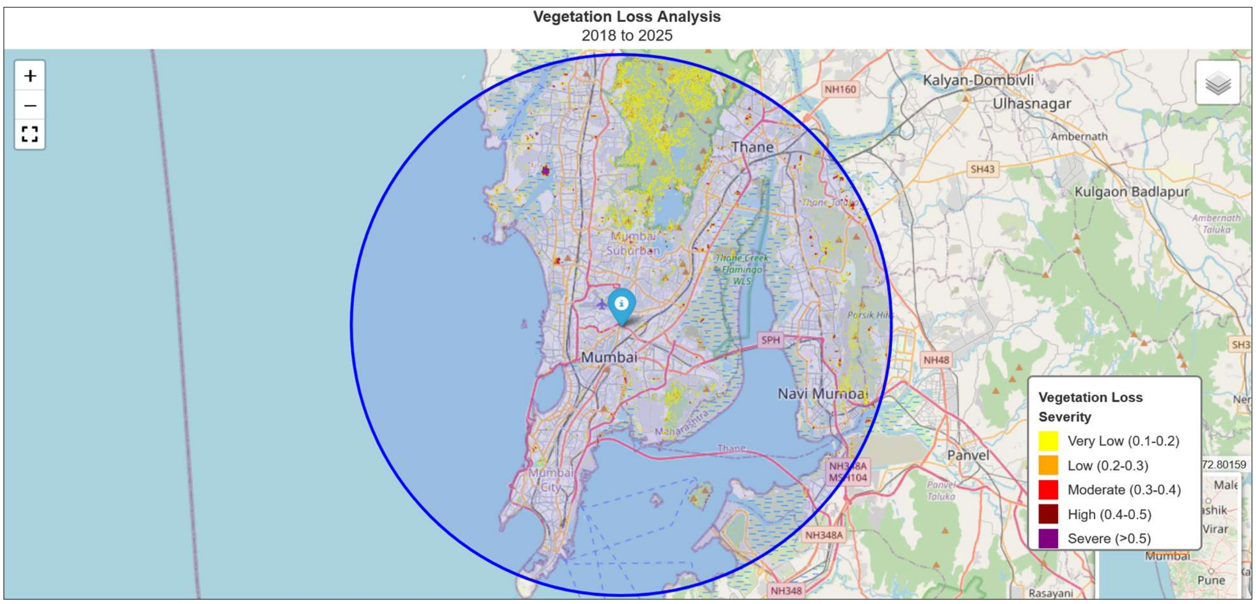


Figure 7.1 Screenshot of Sample Input

7.2.3 Interpretation

- The web tool successfully identified deforested areas even at a fine spatial scale (10 m resolution).

- The interactive map simplifies result interpretation for non-technical users.
- It offers practical utility for planners, educators, and field investigators.

7.3 Multi-Year NDVI Analysis for Mumbai Region (QGIS)

This component focused on year-wise vegetation loss analysis from 2016 to 2024 in the Mumbai Metropolitan Region, using GEE for data processing and QGIS for visualization.

7.3.1 NDVI Time-Series Results

- NDVI composites were generated for each year.
- Reference year: **2024**
- Comparison revealed a consistent pattern of vegetation degradation, especially in peri-urban areas.

7.3.2 Classified Output Maps

- Each yearly comparison resulted in a binary deforestation map exported as a GeoTIFF.
- These maps were visualized in QGIS with red tiles representing areas of confirmed deforestation (based on NDVI drop > 0.1).
- True color composites supported visual verification (ground truthing).

7.3.3 Area Statistics

Vegetation loss was quantified for each year using pixel area calculations. The table below summarizes estimated deforestation for select years:

Comparison	Area in Square Kilometer	Area in Hectares
2016_vs_2024	0.01642	1.642015
2017_vs_2024	0.01642	1.642015

2018_vs_2024	51.1222	5112.22
2019_vs_2024	12.54554	1254.554
2020_vs_2024	22.07787	2207.787
2021_vs_2024	14.37609	1437.609
2022_vs_2024	16.03124	1603.124
2023_vs_2024	8.79067	879.067

Table 7.1 Statistics of Deforestation using Sentinel 2

Comparison	**Area in Square Kilometer**	**Area in Hectares**
2016_vs_2024	47.20863	4720.863
2017_vs_2024	21.00183	2100.183
2018_vs_2024	85.5667	8556.67
2019_vs_2024	12.31376	1231.376
2020_vs_2024	69.02411	6902.411
2021_vs_2024	177.5508	17755.08
2022_vs_2024	106.1619	10616.19
2023_vs_2024	48.4813	4848.13

Table 7.2 Statistics of Deforestation using Landsat 8

Note: Values are approximations derived from GEE area reduction algorithms.

7.3.4 Observations

- The **maximum vegetation loss** was observed in the early years of the time series, with the **2016 vs 2024** comparison showing the **highest NDVI decrease**, confirming long-term vegetation degradation in the Mumbai Metropolitan Region.

- The majority of vegetation loss occurred in **fringe areas and semi-urban belts**, where green patches present in 2016 were significantly reduced or completely absent in recent imagery.
- The red-tiled regions highlighted in QGIS coincided with **known land-use changes**, especially in **eastern Thane**, **Panvel outskirts**, and **peripheral Navi Mumbai**, indicating loss due to expanding infrastructure, commercial zones, and transport corridors.
- Year-on-year comparisons showed **less dramatic loss in recent years (2020–2022)**, suggesting that **most deforestation occurred before or around 2018** [12], with urban saturation reducing the pace in recent years.
- Validation using **true color composites** from Sentinel-2 confirmed that NDVI-classified deforested regions corresponded to visible vegetation decline, lending credibility to the NDVI-based threshold classification used.

7.4 Visual Results

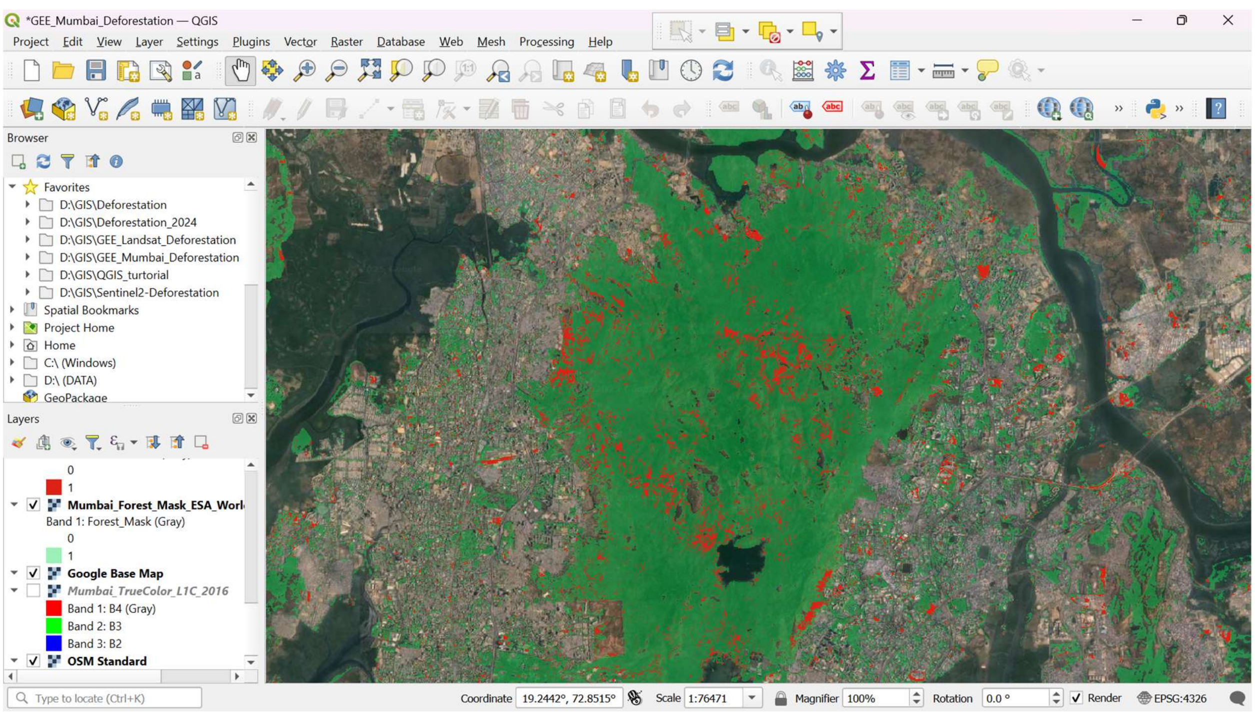


Figure 7.2 Deforestation Map using Sentinel 2.

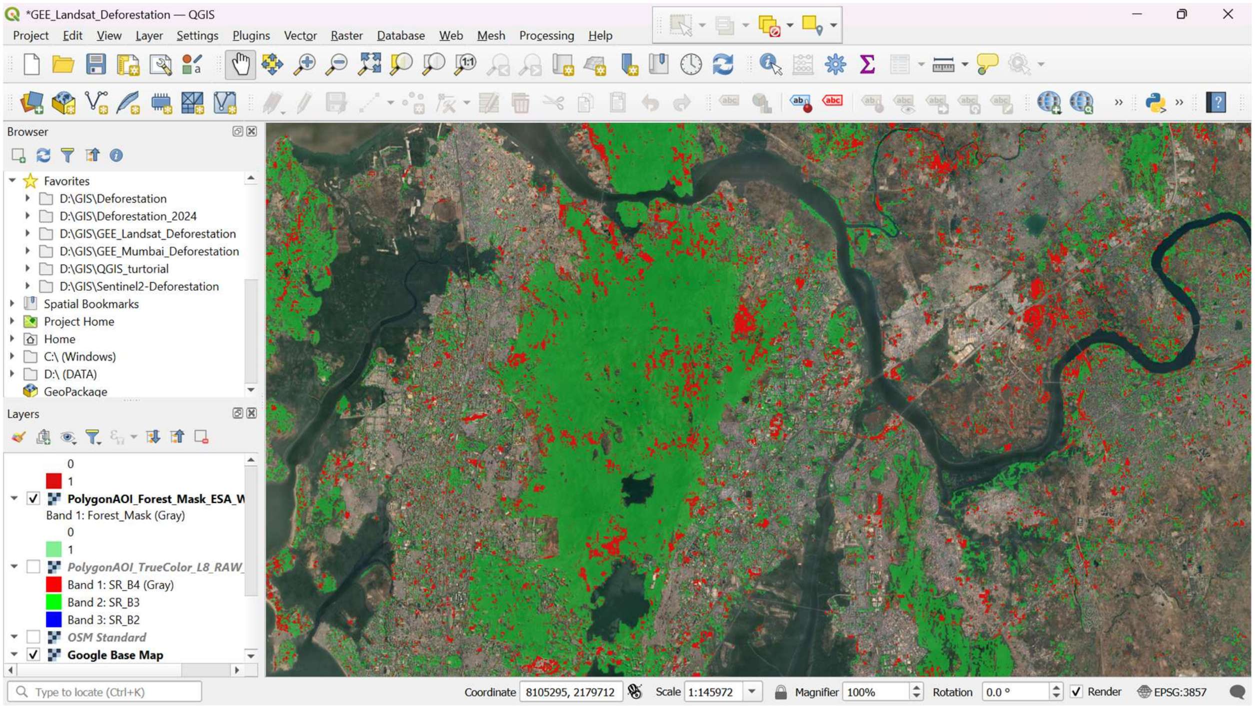


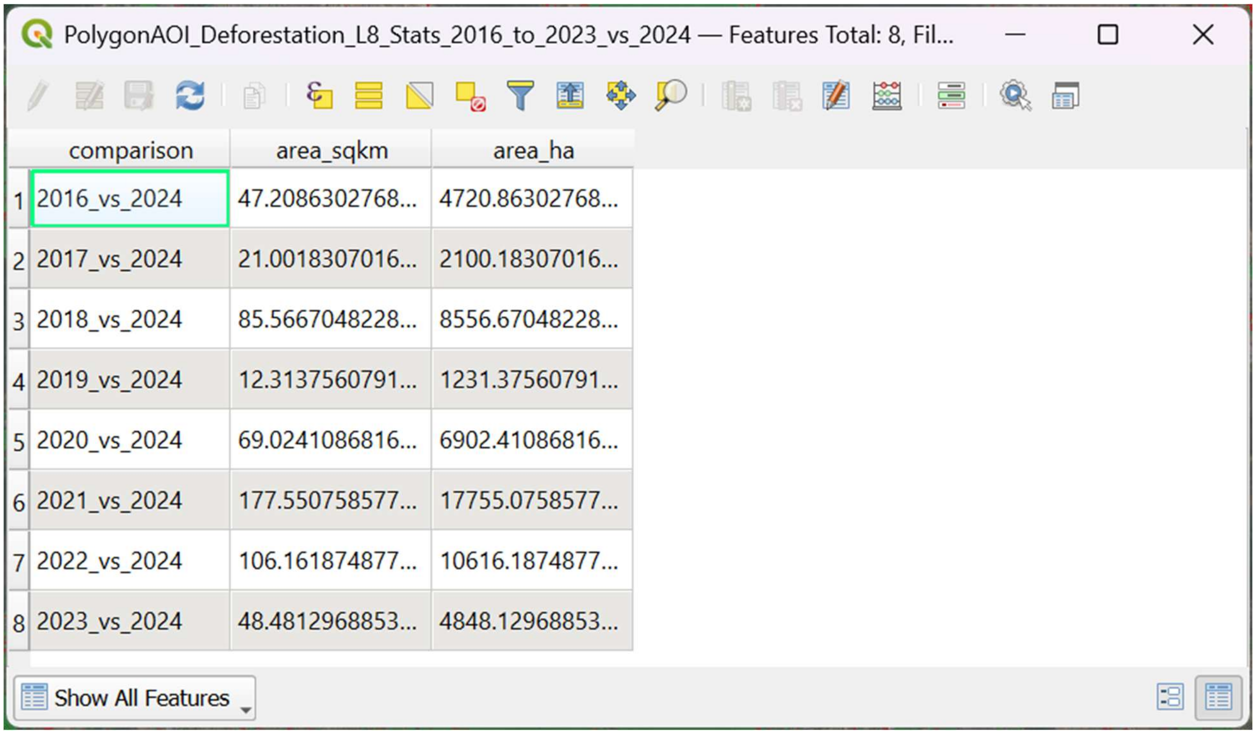
Figure 7.3 Deforestation Map using Landsat 8.

Mumbai_Deforestation_Statistics_2016_to_2023_vs_2024 — Features Total: 8, Filter...

	comparison	area_sqkm	area_ha
1	2016_vs_2024	0.01642015469...	1.64201546936...
2	2017_vs_2024	0.01642015469...	1.64201546936...
3	2018_vs_2024	51.1221983112...	5112.21983112...
4	2019_vs_2024	12.5455420625...	1254.55420625...
5	2020_vs_2024	22.0778651822...	2207.78651822...
6	2021_vs_2024	14.3760877261...	1437.60877261...
7	2022_vs_2024	16.0312400012...	1603.12400012...
8	2023_vs_2024	8.79066999138...	879.066999138...

Show All Features

Figure 7.4 Statistics table of Deforestation using Sentinel 2



PolygonAOI_Deforestation_L8_Stats_2016_to_2023_vs_2024 — Features Total: 8, Fil...

	comparison	area_sqkm	area_ha
1	2016_vs_2024	47.2086302768...	4720.86302768...
2	2017_vs_2024	21.0018307016...	2100.18307016...
3	2018_vs_2024	85.5667048228...	8556.67048228...
4	2019_vs_2024	12.3137560791...	1231.37560791...
5	2020_vs_2024	69.0241086816...	6902.41086816...
6	2021_vs_2024	177.550758577...	17755.0758577...
7	2022_vs_2024	106.161874877...	10616.1874877...
8	2023_vs_2024	48.4812968853...	4848.12968853...

Show All Features

Figure 7.5 Statistics table of Deforestation using Landsat – 8

7.5 Visual Comparison (Groundtruth)

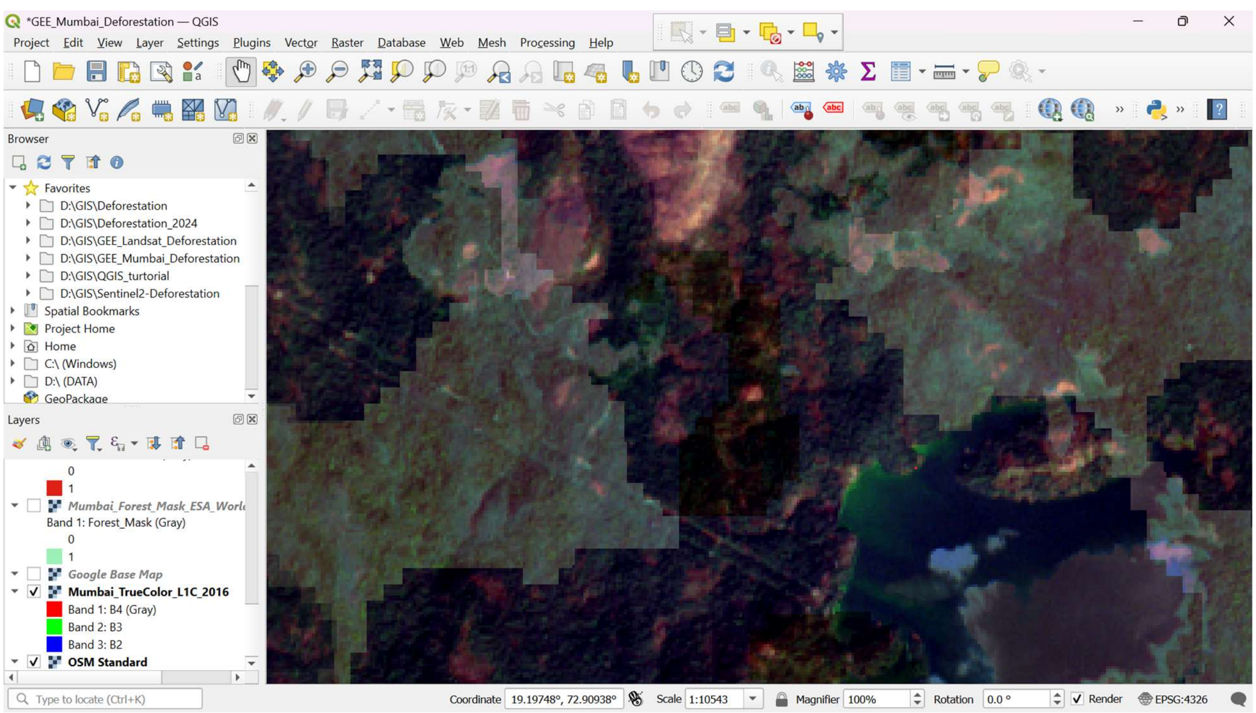


Figure 7.6 – Sentinel-2 True Color Composite of the Region (2016)

This image shows the actual vegetation and forest cover present in the selected region during the year 2016, captured using cloud-free Sentinel-2 imagery. It serves as the base reference for change detection.

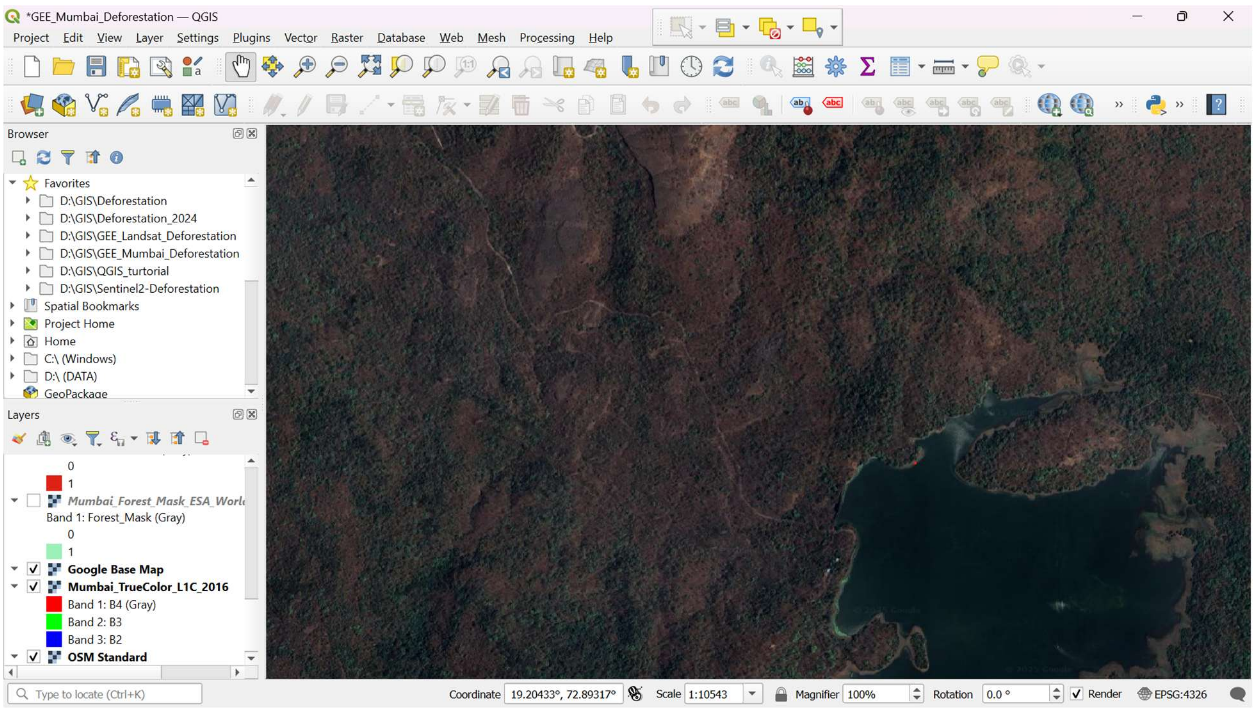


Figure 7.7 – Present-Day View of the Same Location (2025) from Google Earth

This image presents the current state of the region using high-resolution satellite imagery from Google Earth, revealing visible land-use transformation and reduction in vegetation cover.

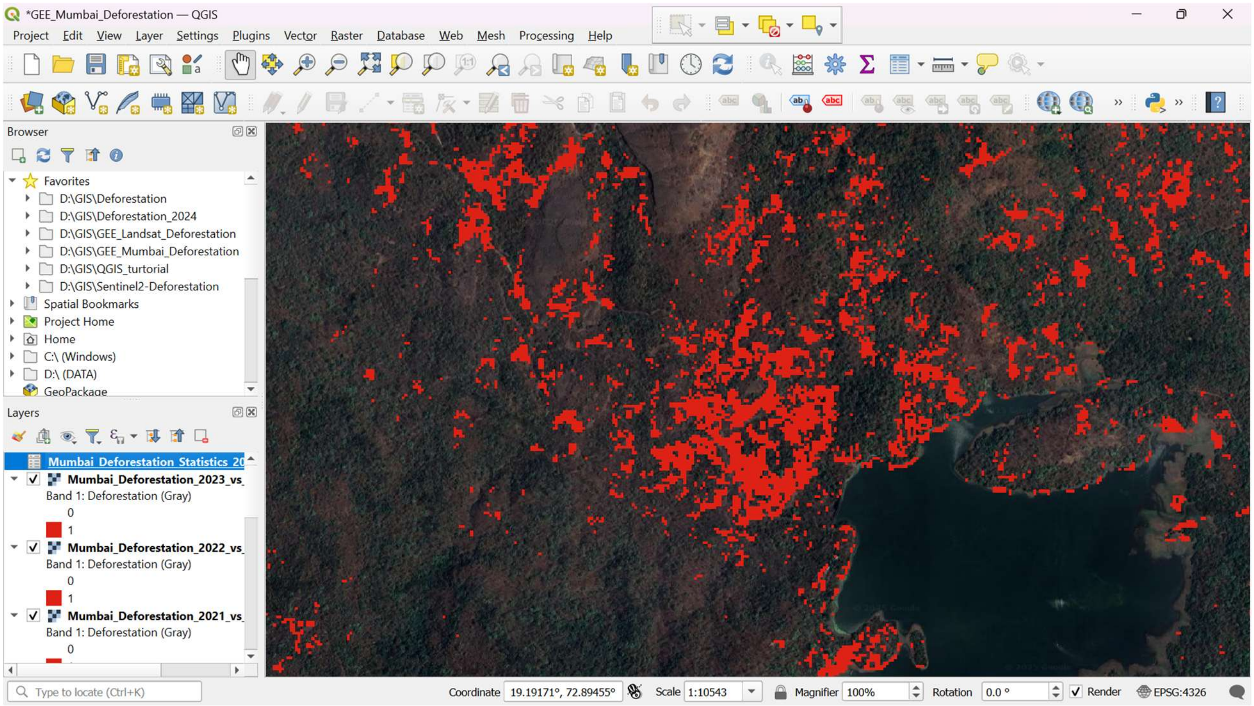


Figure 7.8 – NDVI-Based Deforestation Map Showing Red Tiles Overlay (2016 vs 2024)

This image displays the result of the NDVI loss classification for the same region, where red tiles represent deforested areas detected between 2016 and 2024. The highlighted zones match exactly with visibly degraded patches observed in Google Earth.

Ground truth Conclusion

The ground truth validation confirms the accuracy of NDVI-based deforestation detection. By comparing the true color composite from 2016 with recent high-resolution imagery from Google Earth, and overlaying it with NDVI loss maps, it is evident that the red-tiled deforestation zones correctly align with actual areas of vegetation removal. This alignment strengthens the reliability of the system's classification thresholds and validates the effectiveness of the automated detection methodology used in this project.

7.6 Summary of Results

Module	Outputs Generated	Tools Used	Purpose
Web App	Interactive NDVI loss maps (HTML)	Flask, Folium, GEE	Custom user-driven analysis
Regional Study	GeoTIFFs, CSV stats, QGIS maps	GEE, QGIS	Multi-year urban deforestation monitoring

Table 7.2 Summary of Result Analysis

The results affirm that the system is capable of accurately detecting and visualizing deforestation trends using NDVI analysis. The outputs are both technically robust and user-friendly, achieving the project's goal of accessibility and precision.

8. CONCLUSION AND FUTURE SCOPE

8.1 Conclusion

The Deforestation Monitoring Tool developed in this project effectively demonstrates the integration of Remote Sensing, GIS, and web technologies for monitoring vegetation loss over time. The tool is structured into two main modules:

1. **Web-Based Application** – Enables users to perform real-time, location-specific NDVI-based vegetation loss detection using Google Earth Engine and view results on an interactive Folium map.
2. **Regional Time-Series Analysis** – Focused on the Mumbai Metropolitan Region, this module processes and compares NDVI data from 2016 to 2024, generating GeoTIFF maps and statistical reports, which are analyzed using QGIS.

The system achieves its primary objectives of detecting and visualizing deforestation in a user-friendly, spatially accurate, and efficient manner. By abstracting the complexities of remote sensing data processing, the tool makes environmental monitoring more accessible to non-technical users, researchers, and urban planners.

8.2 Future Scope

While the current system is robust and functional, the following improvements are planned to enhance usability and expand analytical capabilities:

1. **Frontend Enhancement with React.js**
   Migrating the current Flask-rendered map interface to a modern React.js frontend will improve responsiveness, user experience, and provide better control over map layers and visualization components.
2. **Downloadable Report Generation**

   Adding functionality for users to download analysis results as PDF or CSV reports will improve documentation and make the tool more useful for reporting and academic or planning purposes.

3. **Support for Custom Polygon Uploads**

   Allowing users to upload shapefiles or draw custom polygons will expand the system's applicability beyond point-radius inputs and support more flexible, real-world analysis scenarios.

8.3 Final Remarks

The Deforestation Monitoring Tool contributes toward addressing global environmental challenges by offering a scalable and practical system for detecting vegetation loss. It promotes transparency, environmental awareness, and data-driven decision-making among stakeholders. With further development, this system holds the potential to become a valuable asset for smart city governance, ecological research, forest conservation, and sustainability initiatives.

References

[1] C. J. Tucker, “Red and photographic infrared linear combinations for monitoring vegetation,” *Remote Sensing of Environment*, vol. 8, pp. 127–150, 1979.

(Original source of the NDVI formula.)

[2] M. C. Hansen et al., “High-resolution global maps of 21st-century forest cover change,” *Science*, vol. 342, no. 6160, pp. 850–853, Nov. 2013.

(Used in Global Forest Watch – key research in global deforestation analysis.)

[3] N. Gorelick et al., “Google Earth Engine: Planetary-scale geospatial analysis for everyone,” *Remote Sensing of Environment*, vol. 202, pp. 18–27, 2017.

(Describes the capabilities and cloud-computing power of GEE.)

[4] Q. Weng, *Remote Sensing and GIS Integration: Theories, Methods, and Applications*. New York, NY, USA: McGraw-Hill, 2010.

(Describes how RS and GIS are integrated in environmental monitoring.)

[5] European Space Agency (ESA), “Copernicus Sentinel-2 Mission,” 2022. [Online]. Available: https://sentinel.esa.int/web/sentinel/missions/sentinel-2

(Source for Sentinel-2 satellite data used in NDVI analysis.)

[6] United States Geological Survey (USGS), “Landsat 8 Overview,” 2022. [Online]. Available: https://www.usgs.gov/landsat-missions/landsat-8

(Source of Landsat-8 satellite imagery.)

[7] Open Source Geospatial Foundation, “QGIS: A Free and Open Source Geographic Information System,” 2022. [Online]. Available: https://qgis.org/

(GIS software used for post-processing and visualization.)

[8] Python Software Foundation, “Flask Documentation,” 2022. [Online]. Available: https://flask.palletsprojects.com/

(Backend framework used for API and server-side logic.)

[9] Python Software Foundation, “Folium: Python Data, Leaflet.js Maps,” [Online]. Available: https://python-visualization.github.io/folium/

(Used for rendering interactive maps.)

[10] Global Forest Watch, “Monitoring Forests in Near Real-Time,” 2022. [Online]. Available: https://www.globalforestwatch.org/

(Example of global-scale forest monitoring tool using satellite data.)

[11] NASA FIRMS, “Fire Information for Resource Management System,” [Online]. Available: https://firms.modaps.eosdis.nasa.gov/

(Large-scale monitoring platform using MODIS data.)

[12] D. P. Roy et al., “Landsat-8: Science and product vision for terrestrial global change research,” *Remote Sensing of Environment*, vol. 145, pp. 154–172, 2014.

(Scientific background on Landsat 8 use in vegetation and land change detection.)

Appendices

Appendix A: Google Earth Engine Code for Deforestation Detection Using Sentinel-2

```
// Deforestation Mapping for Mumbai Region
// Goal: Detect NDVI loss within forest areas between a start and reference year.
// Outputs: GeoTIFFs for QGIS, CSV statistics.

// Define Area of Interest (AOI) and Time Period
var area = ee.Geometry.Polygon(
    [[[72.75, 18.90],
      [73.25, 18.90],
      [73.25, 19.50],
      [72.75, 19.50],
      [72.75, 18.90]]]);
var startYear = 2016;
var referenceYear = 2024;

// Select Data Sources
var s2Sr = ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED").filterBounds(area);
var worldCover = ee.ImageCollection("ESA/WorldCover/v100").first().clip(area);

// Preprocessing Functions
function maskS2clouds(image) {
  var qa = image.select('QA60');
  var cloudBitMask = 1 << 10;
  var cirrusBitMask = 1 << 11;
  var mask = qa.bitwiseAnd(cloudBitMask).eq(0).and(qa.bitwiseAnd(cirrusBitMask).eq(0));
  var maskValidSR = image.select('B2').gt(0).and(image.select('B2').lt(4000));
  return image.updateMask(mask).updateMask(maskValidSR)
      .select("B2", "B3", "B4", "B8")
      .copyProperties(image, ["system:time_start"]);
}

// Calculate NDVI
function calculateNDVI(image) {
  var ndvi = image.normalizedDifference(['B8', 'B4']).rename('NDVI');
  return image.addBands(ndvi).select('NDVI');
}

// Create Annual Composite
function createAnnualNdviComposite(year) {
  var startDate = ee.Date.fromYMD(year, 1, 1);
  var endDate = ee.Date.fromYMD(year, 12, 31);
  var yearCollection = s2Sr.filterDate(startDate, endDate).map(maskS2clouds);
  var ndviCollection = yearCollection.map(calculateNDVI);
  var medianNdvi = ndviCollection.median().setDefaultProjection('EPSG:4326', null, 10);
```

```
  var count = yearCollection.size();
  var defaultImage = ee.Image.constant(0).rename('NDVI').clip(area).setDefaultProjection('EPSG:4326', null, 10);
  return ee.Algorithms.If(
    count.gt(0),
    ee.Algorithms.If(medianNdvi.bandNames().contains('NDVI'), medianNdvi, defaultImage),
    defaultImage
  );
}

// Identify Forest Areas
var forestMask = worldCover.select('Map').eq(10).rename('Forest_Mask');
Export.image.toDrive({
  image: forestMask.unmask(0).byte(),
  description: 'Mumbai_Forest_Mask_ESA_WorldCover',
  folder: 'GEE_Mumbai_Deforestation',
  scale: 10,
  region: area,
  fileFormat: 'GeoTIFF',
  maxPixels: 1e10
});

// Pre-calculate Reference Year NDVI
var ndviReference = ee.Image(createAnnualNdviComposite(referenceYear)).clip(area);

// Export True Color Reference Image
var startDateGT = ee.Date.fromYMD(startYear, 1, 1);
var endDateGT = ee.Date.fromYMD(startYear, 12, 31);
var trueColorGT = s2Sr.filterDate(startDateGT, endDateGT).map(maskS2clouds)
                  .select(['B4', 'B3', 'B2']).median()
                  .visualize({min: 100, max: 3500, gamma: 1.4});
Export.image.toDrive({
  image: trueColorGT,
  description: 'Mumbai_TrueColor_' + startYear,
  folder: 'GEE_Mumbai_Deforestation',
  scale: 10,
  region: area,
  fileFormat: 'GeoTIFF',
  maxPixels: 1e10
});

// Loop Through Years: Detect Change, Calculate Stats, Export
var deforestationStats = [];
var ndviLossThreshold = 0.15;
for (var year = startYear; year < referenceYear; year++) {
  var ndviYear = ee.Image(createAnnualNdviComposite(year)).clip(area);
  var ndviDifference = ndviReference.subtract(ndviYear).rename('NDVI_Change');
  var potentialLoss = ndviDifference.lt(-ndviLossThreshold).rename('Potential_Loss');
  var deforestation = potentialLoss.updateMask(forestMask).rename('Deforestation');
```

```
  var pixelArea = ee.Image.pixelArea();
  var deforestationAreaImage = deforestation.multiply(pixelArea);
  var stats = deforestationAreaImage.reduceRegion({
    reducer: ee.Reducer.sum(),
    geometry: area,
    scale: 10,
    maxPixels: 1e10,
    tileScale: 4
  });
  var rawArea = stats.get('Deforestation');
  var areaSqM = ee.Algorithms.If(rawArea, ee.Number(rawArea), ee.Number(0));
  var areaSqKm = ee.Number(areaSqM).divide(1e6);
  var areaHa = ee.Number(areaSqM).divide(1e4);
  var comparisonLabel = year + '_vs_' + referenceYear;
  deforestationStats.push({
    'comparison': comparisonLabel,
    'area_sqkm': areaSqKm,
    'area_ha': areaHa
  });
  Export.image.toDrive({
    image: deforestation.unmask(0).byte(),
    description: 'Mumbai_Deforestation_' + comparisonLabel,
    folder: 'GEE_Mumbai_Deforestation',
    scale: 10,
    region: area,
    fileFormat: 'GeoTIFF',
    maxPixels: 1e13
  });
}

// Export Statistics Table
var features = ee.FeatureCollection(deforestationStats.map(function(dict) {
  return ee.Feature(null, dict);
}));
Export.table.toDrive({
  collection: features,
  description: 'Mumbai_Deforestation_Statistics_' + startYear + '_to_' + (referenceYear - 1) + '_vs_' + referenceYear,
  folder: 'GEE_Mumbai_Deforestation',
  fileFormat: 'CSV',
  selectors: ['comparison', 'area_sqkm', 'area_ha']
});
```

Appendix B: Google Earth Engine Code for Deforestation Detection Using Landsat-8

```
// Deforestation Mapping Using Landsat-8
// Goal: Detect NDVI loss within forest areas using Landsat-8 SR data.
// Outputs: GeoTIFFs for QGIS, CSV statistics.

// Define Area of Interest (AOI)
var area = ee.Geometry.Polygon(
    [[[72.75, 18.90],
      [73.25, 18.90],
      [73.25, 19.50],
      [72.75, 19.50],
      [72.75, 18.90]]]);
var startYear = 2016;
var referenceYear = 2024;

// Select Data Sources
var l8sr = ee.ImageCollection("LANDSAT/LC08/C02/T1_L2").filterBounds(area);
var worldCover = ee.ImageCollection("ESA/WorldCover/v100").first().clip(area);

// Preprocessing Functions
function maskL8sr(image) {
  var qa = image.select('QA_PIXEL');
  var clearMask = qa.bitwiseAnd(1 << 1).eq(0)
                  .and(qa.bitwiseAnd(1 << 2).eq(0))
                  .and(qa.bitwiseAnd(1 << 3).eq(0))
                  .and(qa.bitwiseAnd(1 << 4).eq(0));
  var opticalBands = image.select('SR_B.').multiply(0.0000275).add(-0.2);
  return image.addBands(opticalBands, null, true)
              .updateMask(clearMask)
              .select('SR_B1', 'SR_B2', 'SR_B3', 'SR_B4', 'SR_B5', 'SR_B6', 'SR_B7')
              .copyProperties(image, ["system:time_start"]);
}

// Calculate NDVI
function calculateL8Ndvi(image) {
  var ndvi = image.normalizedDifference(['SR_B5', 'SR_B4']).rename('NDVI');
  return image.addBands(ndvi).select('NDVI');
}

// Create Annual Composite
function createAnnualL8NdviComposite(year) {
  var startDate = ee.Date.fromYMD(year, 1, 1);
  var endDate = ee.Date.fromYMD(year, 12, 31);
  var yearCollection = l8sr.filterDate(startDate, endDate).map(maskL8sr);
  var ndviCollection = yearCollection.map(calculateL8Ndvi);
  var medianNdvi = ndviCollection.median().setDefaultProjection('EPSG:4326', null, 30);
```

```
  var count = yearCollection.size();
  var defaultImage = ee.Image.constant(0).rename('NDVI').clip(area).setDefaultProjection('EPSG:4326', null, 30);
  return ee.Algorithms.If(
    count.gt(0),
    ee.Algorithms.If(medianNdvi.bandNames().contains('NDVI'), medianNdvi, defaultImage),
    defaultImage
  );
}

// Identify Forest Areas
var forestMask = worldCover.select('Map').eq(10).rename('Forest_Mask');
Export.image.toDrive({
  image: forestMask.unmask(0).byte(),
  description: 'PolygonAOI_Forest_Mask_ESA_WorldCover',
  folder: 'GEE_Landsat_Deforestation_Output',
  scale: 10,
  region: area,
  fileFormat: 'GeoTIFF',
  maxPixels: 1e10
});

// Pre-calculate Reference Year NDVI
var ndviReferenceL8 = ee.Image(createAnnualL8NdviComposite(referenceYear)).clip(area);

// Export True Color Reference Image
var startDateGT = ee.Date.fromYMD(startYear, 1, 1);
var endDateGT = ee.Date.fromYMD(startYear, 12, 31);
var trueColorCompositeL8 = l8sr.filterDate(startDateGT, endDateGT).map(maskL8sr)
                           .select(['SR_B4', 'SR_B3', 'SR_B2']).median();
Export.image.toDrive({
  image: trueColorCompositeL8.toFloat(),
  description: 'PolygonAOI_TrueColor_L8_RAW_' + startYear,
  folder: 'GEE_Landsat_Deforestation_Output',
  scale: 30,
  region: area,
  fileFormat: 'GeoTIFF',
  maxPixels: 1e10
});

// Loop Through Years: Detect Change, Calculate Stats, Export
var deforestationStatsL8 = [];
var ndviLossThresholdL8 = 0.1;
for (var year = startYear; year < referenceYear; year++) {
  var ndviYearL8 = ee.Image(createAnnualL8NdviComposite(year)).clip(area);
  var ndviDifferenceL8 = ndviReferenceL8.subtract(ndviYearL8).rename('NDVI_Change');
  var potentialLossL8 = ndviDifferenceL8.lt(-ndviLossThresholdL8).rename('Potential_Loss');
  var deforestationL8 = potentialLossL8.updateMask(forestMask).rename('Deforestation');
  var pixelArea = ee.Image.pixelArea();
```

```
  var deforestationAreaImageL8 = deforestationL8.multiply(pixelArea);
  var statsL8 = deforestationAreaImageL8.reduceRegion({
    reducer: ee.Reducer.sum(),
    geometry: area,
    scale: 30,
    maxPixels: 1e10,
    tileScale: 4
  });
  var rawAreaL8 = statsL8.get('Deforestation');
  var areaSqML8 = ee.Algorithms.If(rawAreaL8, ee.Number(rawAreaL8), ee.Number(0));
  var areaSqKmL8 = ee.Number(areaSqML8).divide(1e6);
  var areaHaL8 = ee.Number(areaSqML8).divide(1e4);
  var comparisonLabelL8 = year + '_vs_' + referenceYear;
  deforestationStatsL8.push({
    'comparison': comparisonLabelL8,
    'area_sqkm': areaSqKmL8,
    'area_ha': areaHaL8
  });
  Export.image.toDrive({
    image: deforestationL8.unmask(0).byte(),
    description: 'PolygonAOI_Deforestation_L8_' + comparisonLabelL8,
    folder: 'GEE_Landsat_Deforestation_Output',
    scale: 30,
    region: area,
    fileFormat: 'GeoTIFF',
    maxPixels: 1e13
  });
}

// Export Statistics Table
var featuresL8 = ee.FeatureCollection(deforestationStatsL8.map(function(dict) {
  return ee.Feature(null, dict);
}));
Export.table.toDrive({
  collection: featuresL8,
  description: 'PolygonAOI_Deforestation_L8_Stats_' + startYear + '_to_' + (referenceYear - 1) + '_vs_' +
referenceYear,
  folder: 'GEE_Landsat_Deforestation_Output',
  fileFormat: 'CSV',
  selectors: ['comparison', 'area_sqkm', 'area_ha']
});
```

Appendix C: Google Earth Engine Code for Web App

```
from flask import Flask, request, jsonify, render_template_string
from flask_cors import CORS
import folium
from folium import plugins
import ee
import json
from datetime import datetime
import logging

# Initialize Flask app
app = Flask(__name__)
CORS(app)

# Configure logging
logging.basicConfig(level=logging.INFO)
logger = logging.getLogger(__name__)

# Initialize Google Earth Engine
try:
    ee.Initialize(project='ee-deforest-monitor-tool')
    logger.info("Earth Engine initialized successfully")
except Exception as e:
    logger.error("Earth Engine initialization failed: %s", str(e))
    raise

# Constants
VEGETATION_THRESHOLD = 0.2
CLOUD_FILTER = 60
NDVI_DECREASE_THRESHOLD = 0.1

def mask_clouds(image):
    """Improved cloud masking for Sentinel-2."""
    qa = image.select('QA60')
    cloud_bit_mask = (1 << 10) | (1 << 11)
    mask = qa.bitwiseAnd(cloud_bit_mask).eq(0)
    return image.updateMask(mask)

def calculate_ndvi(image):
    """Calculate NDVI only."""
    try:
        ndvi = image.normalizedDifference(['B8', 'B4']).rename('NDVI')
        return image.addBands(ndvi)
    except Exception as e:
        logger.warning(f"NDVI calculation failed: {str(e)}")
        return image
```

```
def get_year_composite(year, geometry):
    """Create annual composite with proper image handling."""
    try:
        start_date = ee.Date.fromYMD(year, 1, 1)
        end_date = ee.Date.fromYMD(year, 12, 31)
        collection = (ee.ImageCollection('COPERNICUS/S2_SR_HARMONIZED')
                      .filterBounds(geometry)
                      .filterDate(start_date, end_date)
                      .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', CLOUD_FILTER))
                      .select(['B2', 'B3', 'B4', 'B8', 'B12', 'QA60'])
                      .map(mask_clouds)
                      .map(calculate_ndvi))
        collection_size = collection.size().getInfo()
        if collection_size == 0:
            logger.warning(f"No images found for year {year} in the specified region")
            return None
        return collection.median().clip(geometry)
    except Exception as e:
        logger.error(f"Error creating composite for {year}: {str(e)}")
        return None

def generate_vegetation_loss_map(latitude, longitude, distance, base_year):
    """Generate vegetation loss map with robust data handling."""
    try:
        distance_meters = distance * 1000
        roi = ee.Geometry.Point([longitude, latitude]).buffer(distance_meters)
        current_year = datetime.now().year
        base_year_composite = get_year_composite(base_year, roi)
        current_year_composite = get_year_composite(current_year, roi)
        if base_year_composite is None:
            raise ValueError(f"No suitable imagery found for base year {base_year}")
        if current_year_composite is None:
            raise ValueError(f"No suitable imagery found for current year {current_year}")
        ndvi_diff = current_year_composite.select('NDVI').subtract(
            base_year_composite.select('NDVI')).rename('NDVI_diff')
        vegetation_mask = base_year_composite.select('NDVI').gt(VEGETATION_THRESHOLD)
        vegetation_loss = ndvi_diff.lt(-NDVI_DECREASE_THRESHOLD).rename('vegetation_loss')
        masked_loss = vegetation_loss.updateMask(vegetation_mask)
        severity = ee.Image(1)
        severity = severity.where(ndvi_diff.lt(-0.2), 2)
        severity = severity.where(ndvi_diff.lt(-0.3), 3)
        severity = severity.where(ndvi_diff.lt(-0.4), 4)
        severity = severity.where(ndvi_diff.lt(-0.5), 5)
        severity = severity.updateMask(masked_loss).rename('severity')
        return severity, roi
    except Exception as e:
        logger.error(f"Error in vegetation loss map generation: {str(e)}")
        raise
```

```
@app.route('/getData', methods=['GET'])
def get_data():
  try:
    latitude = float(request.args.get('latitude'))
    longitude = float(request.args.get('longitude'))
    distance = float(request.args.get('distance', 10))
    year = int(request.args.get('year', datetime.now().year - 1))
    if not (-90 <= latitude <= 90) or not (-180 <= longitude <= 180):
      return jsonify({'error': 'Invalid coordinates'}), 400
    if distance <= 0 or distance > 1000:
      return jsonify({'error': 'Invalid distance (0-1000 km)'}), 400
    current_year = datetime.now().year
    if year < 2015 or year > current_year:
      return jsonify({'error': f'Invalid year (2015-{current_year})'}), 400
    logger.info(f"Processing request: lat={latitude}, lon={longitude}, distance={distance}km, base_year={year}")
    try:
      severity, roi = generate_vegetation_loss_map(latitude, longitude, distance, year)
    except ValueError as e:
      return jsonify({'error': str(e)}), 400
    vis_params = {
      'min': 1,
      'max': 5,
      'palette': ['yellow', 'orange', 'red', 'darkred', 'purple']
    }
    try:
      map_id_dict = severity.getMapId(vis_params)
      tile_url = map_id_dict['tile_fetcher'].url_format
    except Exception as e:
      logger.error(f"Error generating map tiles: {str(e)}")
      return jsonify({'error': 'Failed to generate map tiles. Please try again.'}), 500
    m = folium.Map(location=[latitude, longitude], zoom_start=10, tiles='OpenStreetMap')
    folium.raster_layers.TileLayer(
      tiles=tile_url,
      attr='Google Earth Engine',
      name=f'Vegetation Loss since {year}',
      overlay=True,
      control=True,
      opacity=0.8,
    ).add_to(m)
    folium.Marker(
      location=[latitude, longitude],
      popup=f"Center Point<br>Lat: {latitude}<br>Lon: {longitude}<br>Year: {year}",
      icon=folium.Icon(color='blue', icon='info-sign')
    ).add_to(m)
    folium.Circle(
      location=[latitude, longitude],
      radius=distance * 1000,
```

```
        color='blue',
        fill=True,
        fill_opacity=0.1,
        popup=f"Study area: {distance} km radius"
    ).add_to(m)
    folium.LayerControl().add_to(m)
    plugins.MiniMap().add_to(m)
    plugins.Fullscreen().add_to(m)
    plugins.MousePosition().add_to(m)
    title_html = '''
        <h3 align="center" style="font-size:16px"><b>Vegetation Loss Analysis</b><br>
        {0} to {1}</h3>
    '''.format(year, current_year)
    m.get_root().html.add_child(folium.Element(title_html))
    legend_html = '''
    <div style="position: fixed;
        bottom: 50px; right: 50px; width: 180px; height: 180px;
        border:2px solid grey; z-index:9999; font-size:14px;
        background-color:white;
        padding: 10px;
        border-radius: 6px;">
        <p style="margin-bottom: 5px;"><b>Vegetation Loss Severity</b></p>
        <div style="display: flex; align-items: center; margin-bottom: 5px;">
            <div style="background-color: yellow; width: 20px; height: 20px; margin-right: 10px;"></div>
            <div>Very Low (0.1-0.2)</div>
        </div>
        <div style="display: flex; align-items: center; margin-bottom: 5px;">
            <div style="background-color: orange; width: 20px; height: 20px; margin-right: 10px;"></div>
            <div>Low (0.2-0.3)</div>
        </div>
        <div style="display: flex; align-items: center; margin-bottom: 5px;">
            <div style="background-color: red; width: 20px; height: 20px; margin-right: 10px;"></div>
            <div>Moderate (0.3-0.4)</div>
        </div>
        <div style="display: flex; align-items: center; margin-bottom: 5px;">
            <div style="background-color: darkred; width: 20px; height: 20px; margin-right: 10px;"></div>
            <div>High (0.4-0.5)</div>
        </div>
        <div style="display: flex; align-items: center;">
            <div style="background-color: purple; width: 20px; height: 20px; margin-right: 10px;"></div>
            <div>Severe (>0.5)</div>
        </div>
    </div>
    '''
    m.get_root().html.add_child(folium.Element(legend_html))
    return render_template_string(m.get_root().render())
except Exception as e:
    logger.error(f"Error processing request: {str(e)}", exc_info=True)
```

```python
        return jsonify({'error': str(e)}), 500

@app.route('/health', methods=['GET'])
def health_check():
    return jsonify({'status': 'healthy'}), 200

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=True)
```

GitHub Link

Deforestation Monitoring Tool