Vegetation Measurement along Line Corridor using Satellite Imagery

A dissertation submitted to the Jawaharlal Nehru Technological University, Hyderabad in partial fulfillment of the requirement for the award of degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

Submitted
by

K. Shriya Reddy (20B81A05A3)

M. Rishitha Reddy (20B81A0592)

S. Pallavi (20B81A0585)

Under the Guidance of
Dr. R. K. Selvakumar
Professor

Department of Computer Science and Engineering

CVR COLLEGE OF ENGINEERING

(An UGC Autonomous Institution, Affiliated to JNTUH, Accredited by NBA, and NAAC)
Vastunagar, Mangalpalli (V), Ibrahimpatnam (M),
Ranga Reddy (Dist.) - 501510, Telangana State

2023-24

CVR COLLEGE OF ENGINEERING

(An UGC Autonomous Institution, Affiliated to JNTUH,
Accredited by NBA, and NAAC)
Vastunagar, Mangalpalli (V), Ibrahimpatnam (M),
Ranga Reddy (Dist.) - 501510, Telangana State.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project work entitled **"Vegetation Measurement along Line Corridor using Satellite Imagery"** is being submitted by **K. Shriya Reddy (20B81A05A3), M. Rishitha Reddy (20B81A0592), and S. Pallavi (20B81A0585)** in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering,** during the academic year 2023-2024.

Project Guide
Dr. R. K. Selvakumar
Professor

Professor-in-charge projects
Dr. G. Balakrishna

External Examiner

Professor and Head, CSE
(Dr. A. Vani Vasthala)

DECLARATION

I hereby declare that this project report titled "**Vegetation Measurement along Line Corridor using Satellite Imagery**" submitted to the Department of Computer Science and Engineering, CVR College of Engineering, is a record of original work done by me under the guidance of **Dr. R. K. Selvakumar**. The information and data given in the report is authentic to the best of my knowledge. This project report is not submitted to any other university or institution for the award of any degree or diploma or published at any time before.

K. Shriya Reddy (20B81A05A3)
M. Rishitha Reddy (20B81A0592)
S. Pallavi (20B81A0585)

Date: May 2024
Place: Hyderabad

ACKNOWLEGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people who made it possible and whose encouragement and guidance has been a source of inspiration throughout the course of the project.

It is a great pleasure to convey our profound sense of gratitude to our principal **Dr. K. Ramamohan Reddy, Dr. A. Vani Vathsala**, Head of CSE Department, CVR College of Engineering, for having been kind enough to arrange necessary facilities for executing the project in the college.

We deem it a pleasure to acknowledge our sense of gratitude to our project guide **Dr. R. K. Selvakumar** under whom we have carried out the project work. Her inclusive and objective guidance and timely advice encouraged us with constant flow of energy to continue the work.

We wish a deep sense of gratitude and heartfelt thanks to management for providing excellent lab facilities and tools. Finally, we thank all those whose guidance helped us in this regard.

ABSTRACT

Our project aims to deploy advanced image classification algorithms for comprehensive monitoring of vegetation and infrastructure monitoring along linear corridors using satellite imagery present a significant challenge in environmental management and infrastructure development. Historically, ground survey methods have been the primary approach for such assessments. However, these methods are often time-consuming, costly, and may lack spatial coverage. In contrast, our research leverages machine learning algorithms and image processing techniques to overcome these limitations.

We propose a comprehensive methodology that integrates pre-processing techniques to enhance image quality, supervised classification for precise delineation of vegetation, and advanced edge detection algorithms to identify electrical lines with improved accuracy. Through rigorous testing, our approach achieves a notable accuracy level of 95.24% in vegetation classification, providing valuable insights for environmental assessments, infrastructure maintenance, and risk mitigation strategies. This research offers a practical framework for stakeholders involved in ecosystem monitoring, biodiversity conservation, and land management practices, facilitating informed decision-making in environmental management and infrastructure development projects along linear corridors.

TABLE OF CONTENTS

LIST OF FIGURES

List OF TABLES

ABBREVIATIONS

1.	CNN	Convolution Neural Network
2.	NDVI	Normalized Difference Vegetation Index
3.	RF	Random Forest
4.	SVM	Support Vector Machine
5.	GB	Gradient Boosting
6.	NIR	Near Infrared
7.	DTM	Digital Terrain Model
8.	UAV	Unmanned Arieal Vehicle
9.	KNN	K Nearest Neighbour
10.	ANN	Artificial Neural Network
11.	HSV	Hue Saturation Value
12.	RGB	Red Green Blue
13.	IP	Image Processing
14.	CED	Canny Edge Detection
15.	HT	Hough Transform
16.	BI	Binary Image
17.	CT	Color Transformation
18.	IS	Image Segmentation
19.	FE	Feature Extraction
20.	ML	Machine Learning

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION

In a world grappling with the escalating threats of climate change, deforestation, and illegal land occupation, the need for efficient and proactive monitoring systems has never been more critical. Our project aims to contribute to environmental protection by developing a robust classification system for distinguishing between vegetation and non-vegetation areas, with a specific focus on differentiating forest and agriculture regions. This initiative is driven by the imperative to combat issues such as forest fires, illegal land use, and the broader impact of climate change on our ecosystems. By leveraging using the technology, we aim to provide government of India with a powerful tool to monitor, protect, and sustain the planet's ecosystems.

1.2 PROBLEM STATEMENT

The monitoring and management of vegetation cover are critical for maintaining ecological balance and addressing environmental challenges such as deforestation, climate change, and natural disasters. Traditional methods of monitoring vegetation cover often rely on labor-intensive field surveys and are limited in their scope and efficiency. To address these challenges, our project aims to deploy advanced image classification algorithms using satellite imagery and aerial data. The primary objective is to develop a comprehensive monitoring system capable of classifying areas as vegetation or non-vegetation, with a focus on distinguishing between forested and agricultural lands. This system will provide environmental agencies, researchers, and policymakers with a powerful tool to monitor, protect, and sustain the planet's ecosystems.

1.3 PROJECT OBJECTIVES

1. Develop and deploy advanced image classification algorithms for monitoring vegetation cover using satellite imagery and aerial data.
2. Classify areas as vegetation or non-vegetation, focusing on distinguishing between forested, agricultural lands, and power lines.
3. Develop a tool for rapid threat detection like forest fires and illegal deforestation, aiding natural habitat preservation, including power line monitoring.

4. Integrate machine learning algorithms to enhance monitoring, detecting changes in vegetation cover, especially distinguishing between forested, agricultural lands, and power line corridors, utilizing multi-spectral and high-resolution imagery.

1.4 PROJECT REPORT ORGANIZATION

There will be a total of 6 chapters in this report. Chapter 1 will introduce the problem statement and motivation, project scope, objective and proposed approach also included. In chapter two, there will be different types of solutions to compare in the literature review. While there is also the system architecture, screenshot for the completed project and class, use case and Activity diagrams in chapter 3. Next chapter 4 will determine the tools to develop the project and the methodology of this project. Besides, in chapter 5 there is the implementation and testing part for the real project. Finally, chapter 6 conclusion which includes is the conclusion, impact, significance, contribution, and future work of this project.

1.5 OVERVIEW

In a world grappling with the escalating threats of climate change, deforestation, illegal land occupation, and power line encroachments, the need for efficient and proactive monitoring systems has never been more critical. Our project aims to contribute to environmental protection by developing a robust classification system for distinguishing between vegetation and non-vegetation areas, with a specific focus on differentiating forest and agriculture regions, as well as detecting power line corridors. This initiative is driven by the imperative to combat issues such as forest fires, illegal land use, encroachment on power line infrastructure, and the broader impact of climate change on our ecosystems.

The primary objective of our project is to deploy advanced technology to create a comprehensive monitoring framework. Through the implementation of image classification algorithms, we intend to analyze satellite imagery and aerial data to classify areas as either vegetation or non-vegetation, including the detection of power line corridors. Further refinement of this classification will enable us to discern between forested and agricultural lands as well as identify power line infrastructure. This system holds immense potential in addressing critical environmental concerns. By identifying and tracking changes in vegetation cover and power line encroachments, we can swiftly respond to and mitigate the impact of forest fires and infrastructure-related incidents. Additionally, the ability to distinguish between forest and agriculture areas aids in detecting unauthorized activities, such as illegal deforestation, land occupation, and encroachments on power line infrastructure, fostering better conservation practices and infrastructure maintenance.

Our project aligns with a broader vision of harnessing technological advancements for environmental sustainability. The proposed classification system not only serves as a vigilant guardian against ecological threats but also offers a valuable tool for monitoring climatic changes and their impact on different land types, including power line corridors. By providing real-time data on vegetation dynamics and power line encroachments, our project seeks to contribute to a more informed and responsive approach to environmental conservation and infrastructure management.

As we delve into the details of our methodology and the technical aspects of the classification algorithms employed, it becomes evident that our work is positioned at the intersection of cutting-edge technology and environmental stewardship. Through this project, we aim to not only develop a robust monitoring system but also contribute meaningfully to the global efforts aimed at preserving our natural habitats, safeguarding critical infrastructure, and combating the adverse effects of climate change.

In the subsequent sections of this report, we will elaborate on the specific methodologies, data sources, and technological frameworks that constitute the backbone of our project, encompassing both vegetation and power line detection. The ultimate goal is to present a comprehensive solution that empowers environmental agencies, infrastructure operators, researchers, and policymakers in their endeavors to protect and sustain our planet's invaluable ecosystems and critical infrastructure networks.

1.6 IMAGE PROCESSING

Image processing involves a series of computational techniques applied to digital images to enhance their quality, extract meaningful information, and derive insights for various applications. Initially, raw images are acquired from sources such as cameras, satellites, or sensors. These images undergo preprocessing steps to correct distortions, reduce noise, and standardize formats. Feature extraction techniques are then employed to identify and quantify relevant patterns or characteristics within the images, such as textures, shapes, or colors. Subsequently, advanced algorithms, including machine learning models like convolutional neural networks or traditional methods like thresholding and filtering, are applied for tasks such as classification, segmentation, or object detection. Post-processing steps refine the results, addressing errors or artifacts and improving overall accuracy. Through this process, image processing enables tasks ranging from medical diagnostics and remote sensing to facial recognition and autonomous navigation. Below Figure 1.1 efficiently depicts the image processing system.

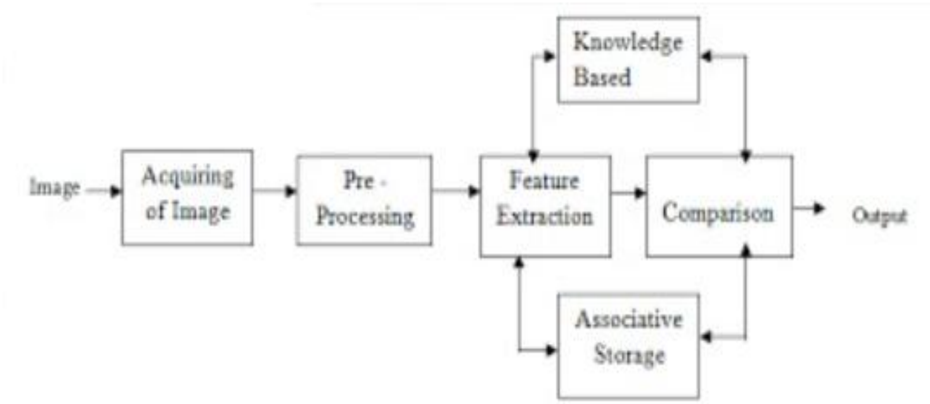


Figure.1.1 Image Processing flow diagram.

a. Data Acquisition:

Image processing plays a pivotal role in data acquisition by enhancing the quality and usability of acquired images. Techniques such as noise reduction, contrast enhancement, and image fusion are employed to improve the clarity and richness of the data obtained from various sources like satellites, cameras, or sensors. By refining raw images, image processing ensures that subsequent analysis is conducted on high-quality data, facilitating more accurate insights and decision-making.

b. Preprocessing:

In preprocessing, image processing techniques are utilized to prepare raw images for analysis by correcting distortions, removing noise, and standardizing formats. Tasks such as geometric and radiometric correction ensure that images are spatially and spectrally consistent, minimizing variability and errors in subsequent processing steps. By enhancing the quality and uniformity of the data, preprocessing optimizes the performance of downstream algorithms, leading to more reliable results in tasks such as classification or feature extraction.

c. **Feature Extraction:**

   Image processing is instrumental in feature extraction, where it identifies and quantifies relevant patterns or characteristics within images. Techniques such as texture analysis, edge detection, and spectral transformation enable the extraction of discriminative features that capture essential information about objects or phenomena depicted in the images. These features serve as the basis for subsequent analysis, facilitating tasks such as object recognition, segmentation, or anomaly detection across diverse domains like medical imaging, remote sensing, and surveillance.

d. **Classification Algorithms:**

   Image processing is at the core of classification algorithms, where it leverages extracted features to categorize images into predefined classes or labels. Supervised learning techniques, including convolutional neural networks (CNNs), support vector machines (SVMs), and decision trees, utilize image features to train models capable of distinguishing between different classes with high accuracy. Through iterative learning and optimization, these algorithms refine their understanding of class boundaries, enabling robust classification of images in applications such as object detection, land cover mapping, and disease diagnosis.

e. **Validation:**

   Image processing aids in the validation of classification results by providing tools and methodologies to assess the accuracy and reliability of the output. Techniques such as confusion matrix analysis, accuracy assessment, and error metrics quantify the performance of classification algorithms by comparing predicted labels with ground truth data. Through visual inspection and statistical evaluation, image processing facilitates the identification of misclassifications, biases, or inconsistencies in the classification output, ensuring the robustness and validity of the analysis results.

Through the integration of advanced image processing techniques, our project not only provides a comprehensive solution for vegetation classification but also sets the stage for a proactive and responsive environmental monitoring system. The subsequent sections will delve into the specifics of our methodology, showcasing the synergy between image processing and environmental conservation in our quest to safeguard the planet's vital ecosystems.

1.7 VIDEO PROCESSING

As part of our comprehensive environmental monitoring system, video processing plays a pivotal role in extending our capabilities beyond static imagery to dynamic surveillance. By harnessing the power of video data, our project aims to provide real-time insights into changes within vegetation and non-vegetation areas, facilitating swift responses to evolving environmental conditions and potential threats. Figure 1.2 in the provided context seems to describe a multimedia database that includes styles, image annotation, video analysis, and employs the MPEG-7 standard for metadata representation using XML.

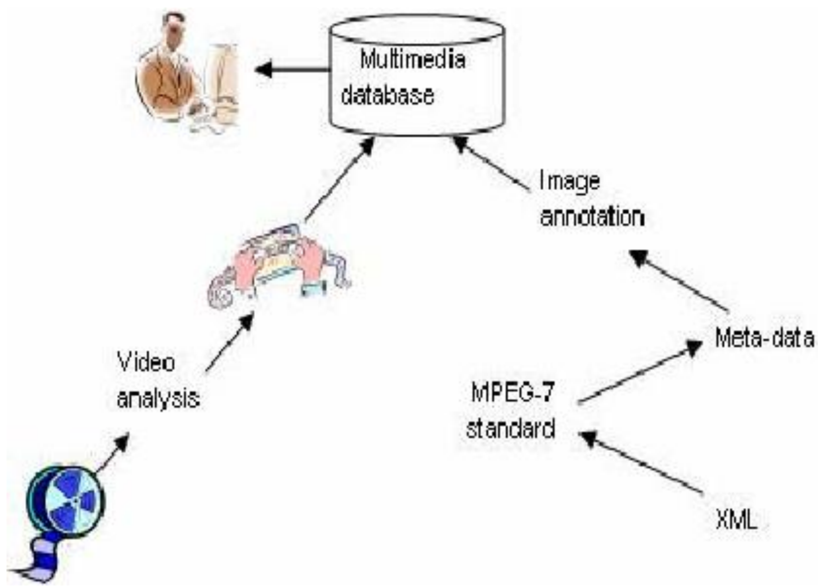


Figure.1.2 Video Processing flow diagram.

a. Data Source:

Video data is sourced from various platforms, including unmanned aerial vehicles (UAVs), fixed surveillance cameras, and other monitoring devices. These sources capture continuous streams of visual information, offering a dynamic perspective of the monitored areas. The temporal dimension introduced by video data enhances our ability to detect and respond to changes over time, such as the rapid progression of a forest fire or unauthorized activities in real-time.

b. Temporal Analysis:

Unlike static imagery, video data introduces the dimension of time, allowing for a nuanced understanding of the environmental dynamics. Temporal analysis is a critical component of our video processing pipeline, enabling the identification of patterns, trends, and anomalies over time. This temporal perspective enhances our capacity to monitor gradual changes, detect sudden events, and assess the impact of climatic variations on vegetation health.

c. Object Tracking:

One of the key challenges in video processing for environmental monitoring is the dynamic nature of the scenes. Object tracking algorithms are employed to follow and trace specific features of interest, such as the spread of a forest fire or the movement of vehicles in protected areas. This facilitates the generation of dynamic maps that depict the evolving status of vegetation and highlight areas of concern.

d. Anomaly Detection:

Anomaly detection algorithms are integrated into our video processing system to identify irregularities or unexpected events within the monitored areas. This includes the early detection of potential threats, such as unauthorized activities, unusual vegetation changes, or signs of impending environmental disasters. The ability to quickly identify anomalies is crucial for initiating timely and targeted responses.

e. Integration with Image Processing:

Video processing is seamlessly integrated with our image processing pipeline to provide a comprehensive monitoring solution. The synergy between static imagery and dynamic video data allows for a holistic understanding of the environmental landscape. This integrated approach enhances the accuracy and reliability of our classification models, offering a unified system for vegetation and land cover analysis.

By incorporating video processing into our environmental monitoring framework, our project not only addresses the challenges posed by dynamic environmental conditions but also lays the foundation for a proactive surveillance system. The subsequent sections will delve into the technical intricacies of our video processing methodology, highlighting its role in creating a responsive and adaptable system for safeguarding our ecosystems.

1.8 COLOR TRANSFORMATION

Color transformation is a fundamental process in image processing that involves converting images from one color space to another. This transformation enables the manipulation and analysis of color information in various applications, such as image enhancement, segmentation, and color correction. By representing colors in different color spaces, such as RGB, YCbCr, or HSV, specific properties of color components, such as brightness, hue, and saturation, can be isolated and manipulated independently. Color transformation facilitates tasks like adjusting color balance, correcting color casts, and enhancing image contrast, contributing to the visual quality and interpretability of images across diverse domains.

RGB Color Space:

The RGB (Red, Green, Blue) color space is a widely used color model in digital imaging, where colors are represented as combinations of red, green, and blue primary colors as shown in Figure 1.3. In RGB, each pixel in an image is represented by three intensity values corresponding to the amount of red, green, and blue light present in that pixel. By varying the intensity of these primary colors, a wide range of colors can be generated, making RGB suitable for displaying images on digital screens. RGB color space is intuitive for capturing and displaying color images, but it may not always correspond well to human perception or be optimal for certain image processing tasks due to its device-dependent nature and lack of perceptual uniformity.

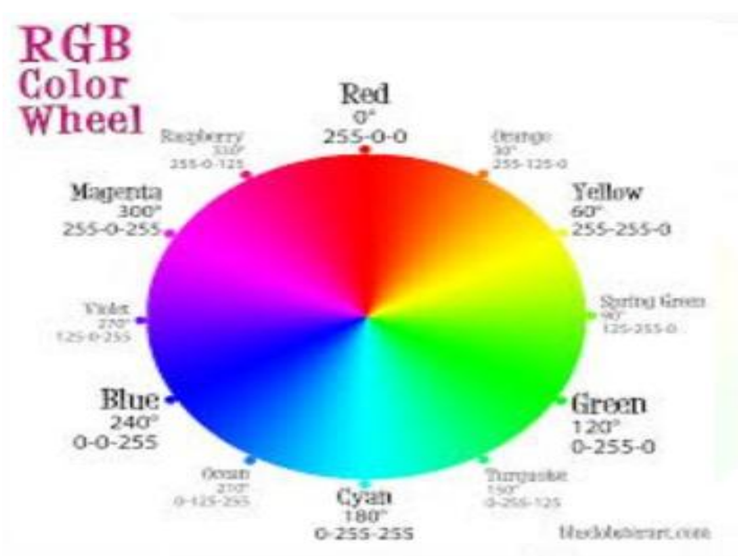


Figure.1.3 RGB Color Space

The below depicts the RGB Representation whose values lie between 0 to 255.

$$\text{Color} = (R, G, B)$$

YCbCr Color Space:

The YCbCr color space separates luminance (Y) from chrominance (Cb and Cr), providing a more perceptually uniform representation of color information compared to RGB. In YCbCr, the Y channel represents brightness or luminance, while the Cb and Cr channels represent color information relative to a reference white point as shown in Figure 1.4. This separation allows for efficient compression of color information in digital media formats such as JPEG, where luminance can be preserved at higher resolution than chrominance, leading to reduced file sizes with minimal perceptual loss. YCbCr is commonly used in video compression, digital television, and image processing applications where efficient representation and manipulation of color information are essential.

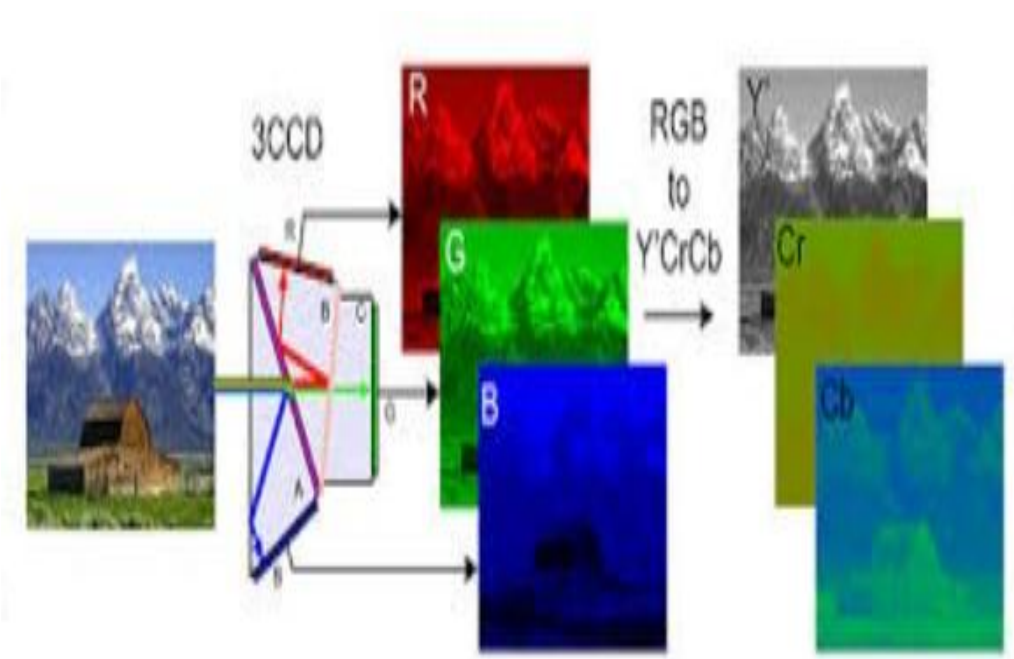


Figure.1.4 YCBR Color Space

The equations for converting RGB to YCbCr is as follows. Equation 1.1 depicts the Luma calculations and equations 1.2 and 1.3 represent Chrominance Blue and Chrominance Red.

- **Y (Luma):**

$$Y = 0.299R + 0.587G + 0.114B \tag{1.1}$$

- **Cb (Chrominance Blue):**

$$Cb = \frac{B - Y}{2(1 - 0.114)} \tag{1.2}$$

- **Cr (Chrominance Red):**

$$Cr = \frac{R - Y}{2(1 - 0.299)} \tag{1.3}$$

HSV Color Space:

The HSV (Hue, Saturation, Value) color space represents colors based on their perceptual attributes of hue, saturation, and brightness. In HSV, hue refers to the dominant wavelength of color, saturation represents the purity or intensity of color, and value corresponds to the brightness or lightness of the color. Unlike RGB, which directly represents colors in terms of primary light sources, HSV organizes colors based on human perception, making it more intuitive for tasks such as color selection, segmentation, and image editing which is shown in Figure 1.6. HSV color space facilitates operations like adjusting color hues, desaturating colors, or isolating specific color ranges, offering greater flexibility and control over color manipulation in image processing applications.

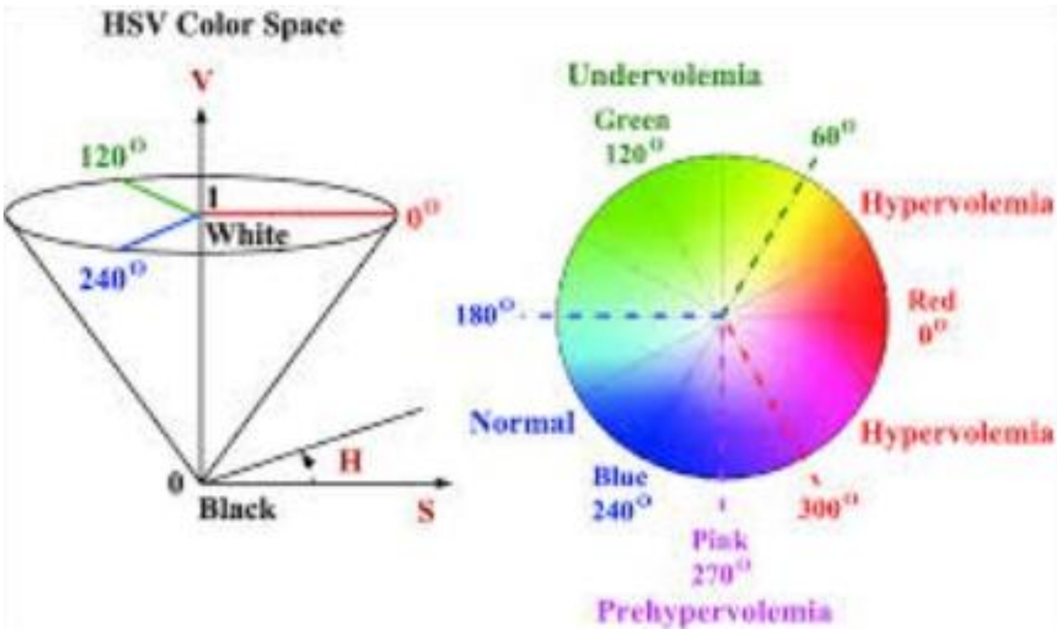


Figure.1.5 HSV Color Space

The equations for converting RGB to HSV is as follows. Equation 1.4 depicts the hue calculations and equations 1.5 and 1.6 represent saturation and intensity.

$$H = \begin{cases} 0 & \text{if } S = 0 \\ \text{mod}\left(\frac{60(G-B)}{S}, 6\right) & \text{if } R = \max(R,G,B) \\ \text{mod}\left(\frac{60(B-R)}{S} + 2, 6\right) & \text{if } G = \max(R,G,B) \\ \text{mod}\left(\frac{60(R-G)}{S} + 4, 6\right) & \text{if } B = \max(R,G,B) \end{cases} \tag{1.4}$$

$$S = \begin{cases} 0 & \text{if } \max(R,G,B) = 0 \\ \frac{\max(R,G,B) - \min(R,G,B)}{\max(R,G,B)} & \text{otherwise} \end{cases} \tag{1.5}$$

$$V = \max(R,G,B) \tag{1.6}$$

1.9 IMAGE ENHANCEMENT

Image enhancement is a set of techniques used to improve the visual quality of an image by increasing its contrast, reducing noise, and enhancing details. This process aims to make images more visually appealing or easier to analyze for specific tasks. Techniques such as contrast stretching, histogram equalization, and spatial filtering are commonly employed to enhance the overall appearance of images while preserving important details and features. Image enhancement finds applications in various fields including medical imaging, surveillance, satellite imaging, and photography, where it plays a crucial role in improving the interpretability and utility of digital images.

Figure.1.6 Image Enhancement

Edge Detection:

Image enhancement plays a critical role in edge detection by improving the clarity and contrast of edges within an image. Enhanced images exhibit better-defined transitions between different intensity levels, making edges more prominent and easier to detect. Techniques such as histogram equalization or contrast stretching can enhance the contrast of edges, making them stand out from the background. This enhanced contrast facilitates the effectiveness of edge detection algorithms, enabling more accurate and reliable identification of object boundaries and structural features within images. Image enhancement thus serves as a preprocessing step in edge detection, enhancing the quality of input images to optimize the performance of subsequent edge detection algorithms.

Histogram Equalization:

Histogram equalization is a technique used to enhance the contrast of an image by redistributing pixel intensities to cover a wider range of values. This process aims to improve the overall brightness and contrast of an image, making it visually more appealing and easier to analyze. Histogram equalization works by transforming the cumulative distribution function of pixel intensities to achieve a more uniform distribution across the intensity range. This results in a stretching of the intensity levels, effectively expanding the dynamic range of the image and enhancing details in both dark and bright regions. Histogram equalization finds applications in various image processing tasks, including medical imaging, satellite imaging, and digital photography, where it helps to improve the visual quality and interpretability of images.

1.10 IMAGE DESCRIPTORS

Region-Based Image Descriptors:

a. Color Histograms:

- Captures the statistical distribution of colors within specific regions of an image.
- Crucial for differentiating between land cover types based on distinct color signatures.
- Contributes to the spectral characterization of the environment.

b. Texture Descriptors (e.g., Local Binary Patterns, GLCM):

- Focuses on spatial patterns and pixel intensity variations within specific regions.
- Essential for capturing textural details in diverse terrain types and vegetation structures.
- Enhances the characterization of regions based on unique textural properties.

Boundary-Based Image Descriptors:

a. Edge and Contour Descriptors (e.g., Sobel, Canny):

- Identifies significant intensity changes, highlighting boundaries between different regions.
- Crucial for delineating features like forest edges and water boundaries.

b. Shape Descriptors (e.g., Hu Moments, Fourier Descriptors):

- Characterizes the geometrical properties of objects, emphasizing shapes within regions.
- Plays a critical role in distinguishing land cover classes based on the shapes of features.
- Contributes to the spatial representation of the environment, focusing on defining shapes.

Additional Image Descriptors:

a. Spectral Descriptors (e.g., NDVI, SAVI):

- Leverages information from different spectral bands to characterize surface reflectance.
- Aids in the accurate classification of vegetation and non-vegetation areas based on spectral signatures.

b. Zernike Moments:

- Represents the shape and spatial distribution of features in an image.
- Focuses on radial and angular information, capturing intricate details within the environment.

By categorizing these image descriptors into region-based and boundary-based types, our environmental monitoring system benefits from a comprehensive set of features. This dual approach ensures that both internal characteristics (regions) and external boundaries are considered, enhancing the overall effectiveness of our classification models.

1.11 SATELLITE IMAGERY

Satellite imagery stands as a remarkable tool, offering us unparalleled perspectives into the multifaceted intricacies that define our planet's landscape. It unveils a breathtaking panorama, unveiling the sprawling expanse of dense forests, the meandering courses of rivers, and the intricate mosaics of urban sprawl. From the verdant canopy of tropical rainforests to the arid expanses of deserts, satellite imagery presents an all-encompassing vista that transcends the confines of borders and boundaries.

In the sphere of environmental monitoring, the adoption of satellite imagery heralds a paradigm shift, ushering in an era marked by transformative insights and proactive stewardship of our ecosystems. By harnessing the power of advanced satellite technology, researchers and conservationists can delve deep into the heart of ecological systems, unraveling complex relationships and identifying critical areas of concern.

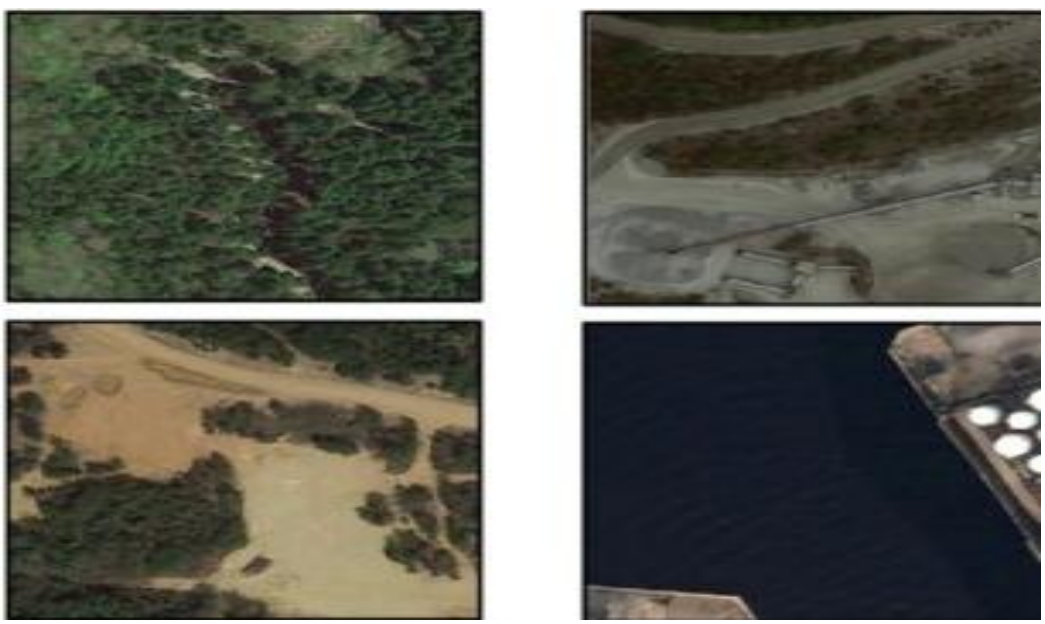


Figure.1.7 Satellite Images

Satellite imagery serves as a sentinel as shown in Figure 1.7. providing real-time data on deforestation, habitat loss, and land use changes. It enables us to track the impact of human activities on fragile ecosystems, facilitating informed decision-making and targeted interventions to mitigate environmental degradation. Moreover, satellite imagery serves as a cornerstone for climate research, offering invaluable insights into the dynamics of weather patterns, sea level rise, and glacier melt.

Beyond its scientific applications, satellite imagery also holds immense value in disaster management and humanitarian efforts. In the wake of natural calamities such as hurricanes, floods, and wildfires, satellite imagery aids in rapid assessment and response, enabling authorities to coordinate relief efforts and allocate resources effectively.

In essence, satellite imagery represents not just a technological marvel, but a gateway to a deeper understanding of our planet's interconnectedness and fragility. It empowers us to embrace a holistic approach to environmental stewardship, one that is rooted in knowledge, compassion, and a steadfast commitment to preserving the rich tapestry of life on Earth.

Global Observations:

One of the defining features of satellite imagery is its ability to offer a global perspective. Satellites in orbit capture images that span continents, oceans, and remote regions, enabling a bird's-eye view of Earth's surface. This expansive coverage is fundamental for monitoring changes on a planetary scale, essential in addressing global environmental challenges.

Temporal Continuity:

Time, an ever-evolving dimension, is captured in the pixels of satellite imagery. These orbiting sentinels provide a temporal continuum, documenting changes over days, seasons, and years. This temporal resolution is pivotal for tracking the ebb and flow of natural processes, assessing the impact of human activities, and responding to environmental shifts in a timely manner.

Remote Sensing Precision:

Satellite imagery is at the forefront of remote sensing technologies, allowing us to observe and measure Earth's surface from afar. This capability is particularly advantageous for studying inaccessible or hazardous areas, contributing to the advancement of environmental studies, climate research, and conservation efforts without the constraints of physical presence.

Spectral Insights:

With sensors capturing data across multiple spectral bands, satellite imagery unveils a spectrum of information beyond what is visible to the naked eye. These spectral insights empower researchers to derive valuable data on vegetation health, land cover types, and environmental parameters, fostering a deeper understanding of ecological dynamics.

Holistic Environmental Monitoring:

The integration of satellite imagery with advanced technologies enables a holistic approach to environmental monitoring. From assessing the health of ecosystems to detecting changes in land use patterns, satellite imagery provides the canvas upon which we paint a comprehensive picture of Earth's environmental health.

As we embark on a journey to monitor, analyze, and protect our planet, the significance of satellite imagery cannot be overstated. It serves as a panoramic lens that transcends geographical, temporal, and even spectral boundaries, offering a holistic view indispensable for informed decision-making, sustainable resource management, and the preservation of the delicate balance that sustains life on Earth.

CHAPTER 2
LITERATURE SURVEY

2.1 EXISTING WORK

2.1.1 Deep learning model for accurate vegetation classification using RGB image only

In this study it aims to enhance the accuracy of Digital Terrain Models (DTMs) by identifying and excluding vegetation, particularly trees, from Digital Surface Models (DSMs). The goal is to employ various inpainting methods on the removed vegetation pixels to reconstruct more precise terrain information and generate a refined DTM. Three DeepLabV3+ models were trained using distinct datasets at different resolutions as shown in Figure 2.1. Notably, the model trained with a dataset closely matching the resolution of the test data demonstrated superior performance. The semantic segmentation results from this model show significant promise for effectively detecting and excluding vegetation in the context of refining DTMs.

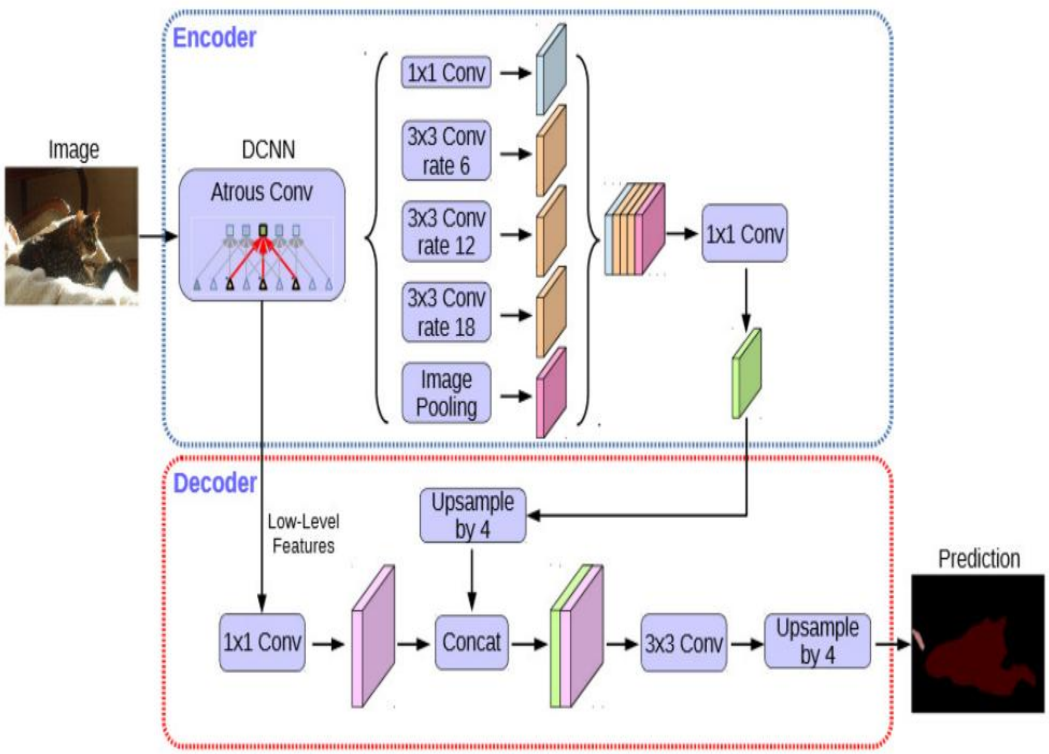


Figure.2.1 Block Diagram of DeepLabV3+

ADVANTAGES:

a. **Accurate Vegetation Classification:**

The use of Deep Learning models, specifically DeepLabV3+, allows for accurate vegetation classification using only RGB images. This approach eliminates the need for additional data sources such as NIR images or LiDAR data, making it cost-effective and accessible.

b. **Digital Terrain Model Improvement:**

The primary objective of the project is to enhance Digital Terrain Models (DTMs) by excluding vegetation from Digital Surface Models (DSMs) based on vegetation types. This targeted approach provides a more accurate representation of the terrain by removing the impact of vegetation.

c. **Inpainting for Terrain Restoration:**

The project employs inpainting methods to restore terrain information by filling in the areas where vegetation has been removed. This can lead to the generation of more accurate DTMs, benefiting applications such as construction surveying, change monitoring, and agricultural management.

d. **Training with Multiple Datasets:**

The use of three different datasets, each collected at different resolutions, demonstrates flexibility and adaptability. The model trained with a dataset having a resolution close to the test data shows the best performance.

e. **Practical Applications:**

The project addresses real-world applications such as emergency landing site selection for unmanned air vehicles (UAVs), agricultural monitoring, and land cover classification for urban planning. These applications have practical implications for various industries.

f. **Low-Cost RGB Image Acquisition:**

The emphasis on using low-cost RGB images, especially from drones equipped with low-cost cameras, makes the approach economically viable.

g. **Comprehensive Analysis:**

The paper provides a detailed analysis of the performance of DeepLabV3+ models trained with different datasets. Performance metrics such as accuracy and mean-intersection-over-union (mIoU) are used for evaluation, offering a comprehensive assessment.

h. **Integration of Field Data:**

   The methodology integrates ground truth data collected through field observations, enhancing the reliability of the classification results. This integration ensures that the model's predictions align with real-world conditions.

DISADVANTAGES:

a. **Resolution Discrepancy:**

   The performance of the model trained with datasets having resolutions different from the test data is suboptimal. This highlights a challenge in generalizing models to scenarios with varying resolutions.

b. **Data Imbalance:**

   Some datasets, like the Slovenia dataset, exhibit data imbalance, where certain land cover types are underrepresented. This can impact the model's ability to accurately classify less frequent classes.

c. **Limited Differentiation:**

   The current models focus on broad vegetation classification (e.g., trees, shrubs, grassland) but do not differentiate between different types of vegetation within these categories (e.g., specific tree species or crop types).

d. **Dependency on Image Characteristics:**

   The project's success is closely tied to the characteristics of the training data, emphasizing the importance of selecting representative datasets. This dependency may limit the model's performance in diverse geographical regions.

e. **Challenges in Urban Areas:**

   While the project addresses vegetation classification, challenges may arise in accurately classifying land cover types in urban areas where multiple non-vegetative elements coexist in close proximity.

f. **Performance Variation:**

   The DeepLabV3+ model trained with the DeepGlobe dataset performs poorly on the Kimisala test images. Understanding the reasons behind such performance variations is crucial for model improvement.

g. **Complexity of Machine Learning Models:**

The adoption of complex deep learning models may pose challenges in terms of computational resources, training time, and model interpretability, especially for users with limited technical expertise.

h. **Need for Further Research:**

The project suggests avenues for future research, such as investigating the model's ability to differentiate between grass, shrub, and tree, and exploring alternative approaches like NDVI-ML and texture features.

2.1.2 Remote sensing-based vegetation classification using machine learning algorithms.

Vegetation plays a crucial role in ecosystems by producing oxygen and absorbing carbon dioxide, creating a habitable environment for humans. Remote sensing is employed in this study to gather essential information about vegetation types. The paper focuses on classifying vegetation using three prominent machine learning algorithms: K-means (an unsupervised classifier), Support Vector Machine (SVM), and Artificial Neural Networks (ANN). The inclusion of K-means allows for a comparison with the supervised classifiers SVM and ANN. Additionally, non-vegetative elements such as buildings, roads, and rivers are categorized. This classification has practical applications, enabling government agencies to obtain crop-specific data (e.g., tobacco, maize) for statistical analysis and mapping. The information is valuable for planning and can contribute to future crop management and urban development strategies.

Methodology description:

In this study, the crucial role of vegetation in ecosystems, contributing to oxygen production and carbon dioxide absorption, is emphasized. Remote sensing is employed to gather essential information about vegetation types. The focus shifts towards classifying vegetation using three prominent machine learning algorithms: K-means (unsupervised), Support Vector Machine (SVM), and Artificial Neural Networks (ANN).

The inclusion of K-means allows for a comprehensive comparison with the supervised classifiers SVM and ANN. The study goes beyond vegetation, categorizing non-vegetative elements such as buildings, roads, and rivers. This classification holds practical applications, providing government agencies with crop-specific data (e.g., tobacco, maize) for statistical analysis and mapping. The information obtained is valuable for planning and can contribute to future crop management and urban development strategies.

A comprehensive methodology is outlined for processing satellite images obtained from the Copernicus database. The process involves three key steps: ground truth data collection and pre-processing, processing, and post-processing. Ground truth data is collected through field observations using a designated app, capturing information about known objects. This data is integrated into the original satellite image, enabling the identification of specific features within the chosen area in Charsadda, Khyber Pakhtunkhwa, Pakistan. Subsequently, the original image is cropped to the area of interest using SNAP software, a process referred to as stacking.

The processing phase in ENVI software involves loading ground truth Regions of Interest (ROIs) onto the stacked image. A mask is created to isolate the desired image, and supervised and unsupervised machine learning algorithms are applied, including artificial neural networks, support vector machines, and K-means clustering. The trained neural network classifies the image into various classes, such as maize, sugarcane, trees, roads, buildings, barren land, bringle, corn, ladyfinger, FCV, and locatt, with unclassified regions for unidentified features.

In the post-processing phase, a confusion matrix is generated to evaluate the accuracy and statistics of the classification results. This matrix provides a comprehensive assessment of the performance of the applied algorithms, allowing for a thorough understanding of the success and limitations of the image classification process.

Overall, this methodology seamlessly integrates field data collection, satellite image processing, and machine learning techniques to derive valuable insights from satellite imagery for the specified region in Pakistan.

ADVANTAGES:

a. **Resource Monitoring:**

   The project can be beneficial for monitoring and managing natural resources. It provides a systematic approach to classifying and understanding the types and distribution of vegetation.

b. **Government Planning:**

   The information obtained, especially regarding crop yields and land cover, can be useful for government agencies in planning and decision-making related to agriculture, town planning, and resource allocation.

c. **Environmental Impact Assessment:**

   The classification of vegetation types and land cover helps in assessing the environmental impact of human activities, such as deforestation and urbanization.

d. **Comparison of Classification Algorithms:**

   The study compares the performance of three machine learning algorithms, shedding light on their effectiveness in classifying vegetation. This can guide future research and applications.

e. **Accuracy Assessment:**

   The confusion matrix and accuracy metrics provide a quantitative assessment of the classification results, allowing for a clear evaluation of the algorithm's performance.

f. **Open Data Usage:**

   Utilizing satellite images from the Copernicus database makes the study replicable and accessible to other researchers, promoting transparency and collaboration in the scientific community.

DISADVANTAGES:

a. **Limited Spatial and Temporal Resolution:**

   Satellite images may have limitations in spatial and temporal resolution, potentially affecting the accuracy of vegetation classification, especially for small or rapidly changing areas.

b. **Dependency on Ground Truth Data:**

   The accuracy of the classification is heavily dependent on the quality of ground truth data collected. Human error or biases during data collection could impact the results.

c. **Algorithm Sensitivity:**

   The choice of machine learning algorithms can significantly impact the results. The study acknowledges the need for choosing suitable algorithms but does not explore the sensitivity of results to algorithm parameters.

d. **Generalization Challenges:**

   The study focuses on a specific region (Charsadda, Pakistan), and the effectiveness of the proposed methods may vary in different geographic locations and environmental conditions.

e. **Data Preprocessing Complexity:**

   The paper briefly mentions preprocessing steps, but the complexity and potential challenges in data preprocessing are not extensively discussed. Preprocessing can significantly influence classification results.

f. **Limited Exploration of Neural Network Architecture:**

   The paper describes using an Artificial Neural Network but lacks detailed information on the architecture, hyperparameter tuning, and considerations for optimal performance.

2.1.3 Vegetation detection using deep learning and conventional methods.

In contemporary land cover analysis, accurate identification of chlorophyll-rich vegetation holds paramount importance across a spectrum of applications ranging from urban growth monitoring to biodiversity conservation, as well as in fields like autonomous navigation and drone mapping. Traditionally, the normalized difference vegetation index (NDVI) has been the cornerstone for vegetation detection in these domains as shown in Figure 2.2. However, this paper embarks on an exploratory journey to evaluate the effectiveness of both conventional methodologies and cutting-edge deep learning techniques in this domain.

The study meticulously scrutinizes two prominent deep learning methodologies, namely DeepLabV3+ and a bespoke convolutional neural network (CNN), to gauge their efficacy in vegetation detection. This evaluation extends across diverse datasets sourced from geographically disparate sites, encompassing varying image resolutions to capture the full spectrum of real-world scenarios. Additionally, the paper introduces an innovative object-based vegetation detection framework that amalgamates NDVI computation with advanced computer vision and machine learning algorithms.

Central to the evaluation is the utilization of high-resolution airborne color images, comprising both RGB and near-infrared (NIR) bands. This choice of imagery enables a comprehensive assessment of the methodologies under consideration, leveraging the rich spectral information inherent in the NIR band. Moreover, the deep learning methodologies are subjected to additional scrutiny by testing them solely with RGB images, thereby simulating scenarios where access to NIR data may be limited.

The paper meticulously examines the detection performance of deep learning methodologies in comparison to the object-based approach, elucidating their respective strengths and limitations. To provide readers with a tangible understanding of the methodologies in action, the paper includes illustrative sample images drawn from the datasets under consideration. Through this comprehensive analysis, the paper endeavors to shed light on the evolving landscape of vegetation detection methodologies, paving the way for informed decision-making and advancement in this.

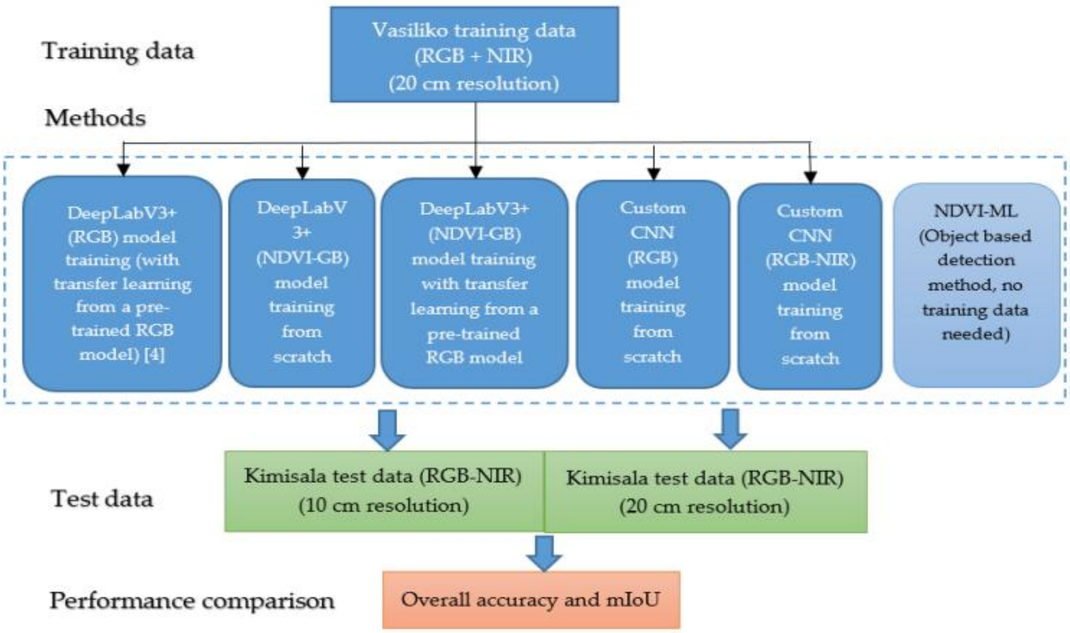


Figure.2.2 Block Diagram of deep learning and conventional methods

ADVANTAGES

a. Improved Detection Accuracy:

- Deep learning methods, especially customized CNN, demonstrate enhanced accuracy in vegetation detection compared to conventional NDVI-based methods.
- Object-based approach (NDVI-ML) outperforms deep learning methods, indicating a potential for more robust detection.

b. Utilization of Multiple Bands:

- Customized CNN's ability to handle both RGB and NIR bands allows for a more comprehensive analysis of vegetation characteristics.
- Integration of NDVI band in DeepLabV3+ training demonstrates the potential to leverage additional spectral information.

c. **Innovative Object-Based Approach:**

- Introduction of NDVI-ML provides a novel and simple method for vegetation detection without the need for extensive training data.
- NDVI-ML outperforms deep learning methods, showcasing the effectiveness of combining NDVI with computer vision and machine learning techniques.

d. **Versatility in Applications:**

- Successful vegetation detection contributes to various applications such as urban growth monitoring, autonomous navigation, and biodiversity conservation.
- The project's findings have implications for diverse fields like agriculture, land-use planning, and environmental monitoring.

DISADVANTAGES:

a. **Challenges in Extending DeepLabV3+:**

- Extending DeepLabV3+ to include NIR bands faces difficulties, limiting the model's ability to fully utilize the available spectral information.
- Architecture modifications and the lack of pre-trained models for additional bands pose significant challenges.

a. **Training Data Limitations:**

- Customized CNN and other deep learning methods require extensive training data, and the absence of pre-trained models for NIR band adds complexity.
- Imbalanced datasets can lead to biased results, especially for underrepresented vegetation classes.

b. **Practicality and Training Starting Point:**

- Training deep learning models from scratch, especially for the NIR band, may not be practical due to the vast amount of required training data.
- DeepLabV3+ training from scratch without pre-trained models hampers its potential performance.

c. **Dataset Specifics:**

- Performance variations across different datasets with varying resolutions and capturing hardware suggest sensitivity to dataset characteristics.

- Transferability of models to new applications might be challenging due to the need for similar characteristics in training and testing data.

d. Limitations of DeepLabV3+:

- DeepLabV3+ only accepts three input channels, requiring modifications for more than three channels, and necessitating training from scratch.
- Existing challenges in handling more than three channels may hinder its applicability to datasets with additional spectral bands.

e. Considerations for Imbalanced Datasets:

- Imbalanced datasets may result in biased loss functions, affecting the classification performance, particularly for underrepresented vegetation classes.
- Careful attention is needed to mitigate issues related to class imbalances in training data.

2.2 LIMITATIONS OF EXISTING WORK

1.K-means Clustering:

- Sensitive to initial centroid placement.
- Influenced by outliers.
- Assumes circular or spherical cluster shapes, limiting adaptability.

2.Support Vector Machines (SVM):

- Sensitivity to noise.
- Reliance on an appropriate kernel function.
- Computational complexity, especially with large datasets.

3.Artificial Neural Networks (ANN):

- Require substantial data for effective training.
- Lack interpretability due to their black-box nature.
- Susceptible to overfitting, particularly with small datasets.
- Computational intensity and the need for careful selection of architecture and hyperparameters.

4. DeepLabV3:

- Notable disadvantage in accurately distinguishing vegetation types.
- especially when using larger backbones or when processing large images, can require significant amounts of memory. This can pose challenges when deploying the model on devices with limited memory resources.

The methodologies of k-means clustering, Support Vector Machines (SVM), and Artificial Neural Networks (ANN) each exhibit distinctive limitations. K-means clustering is sensitive to initial centroid placement, influenced by outliers, and assumes circular or spherical cluster shapes, restricting its adaptability to more complex data structures. Support Vector Machines demonstrate sensitivity to noise, reliance on an appropriate kernel function, and computational complexity, especially with large datasets, posing challenges in scenarios with noisy or extensive data. On the other hand, Artificial Neural Networks demand substantial amounts of data for effective training, lack interpretability due to their black-box nature, and are susceptible to overfitting, particularly in cases of small datasets. Additionally, the computational intensity and the necessity for careful selection of architecture and hyperparameters further complicate the application of neural networks. Despite their effectiveness, these methods necessitate thoughtful consideration of their specific limitations and appropriateness for the characteristics of the data and problem at hand. DeeplabV3 which is used in one of the papers also has a disadvantage as it could not give accurate results while distinguishing vegetation types.

CHAPTER 3

SOFTWARE & HARDWARE SPECIFICATIONS

3.1 SOFTWARE REQUIREMENTS

3.1.1 Functional Requirements

a. Real-time Monitoring: The system should provide real-time monitoring of vegetation cover changes and updates to stakeholders.

b. Security: The system should ensure the security and confidentiality of sensitive data, such as satellite imagery and monitoring reports.

c. Data Integration: The system should integrate data from multiple sources, including satellite imagery, weather data, and ground observations, to provide comprehensive insights.

d. User Interface: The system should have a user-friendly interface for stakeholders to visualize monitoring data and make informed decisions.

e. Scalability: The system should be scalable to handle large volumes of satellite and aerial imagery data.

3.1.2 Non-Functional Requirements

a. Performance: - The system should be able to process and analyze data efficiently to provide timely insights to stakeholders.

b. Reliability: The system should be reliable, with minimal downtime and accurate reporting of monitoring data

c. Usability: The system should have a user-friendly interface that is easy to navigate for stakeholders with varying levels of technical expertise.

d. Accuracy: The classification algorithms should have high accuracy in distinguishing between vegetation and non-vegetation areas, as well as in classifying forested and agricultural lands.

3.2 SYSTEM SPECIFICATIONS

3.2.1 Hardware Specifications

1. Processor-Intel i5 or higher
2. RAM – 16GB or higher
3. Storage-256GB SSD or above
4. Internet connection-40Mbps or more

3.2.2 Software Specifications

1. Platform- any platform that supports python environment
2. Operating system- Windows, Linux or macOS
3. Python- 3 or higher

CHAPTER 4

PROPOSED SYSTEM DESIGN

4.1 PROPOSED SYSTEM: VCML

Our approach utilizes a grid-based segmentation technique to analyse satellite imagery and aerial data for monitoring vegetation cover. The input data, consisting of high-resolution images, is divided into a grid of smaller segments, allowing for more efficient analysis and classification.the below Figure 4.1 demonstrates the architecture of VCML.

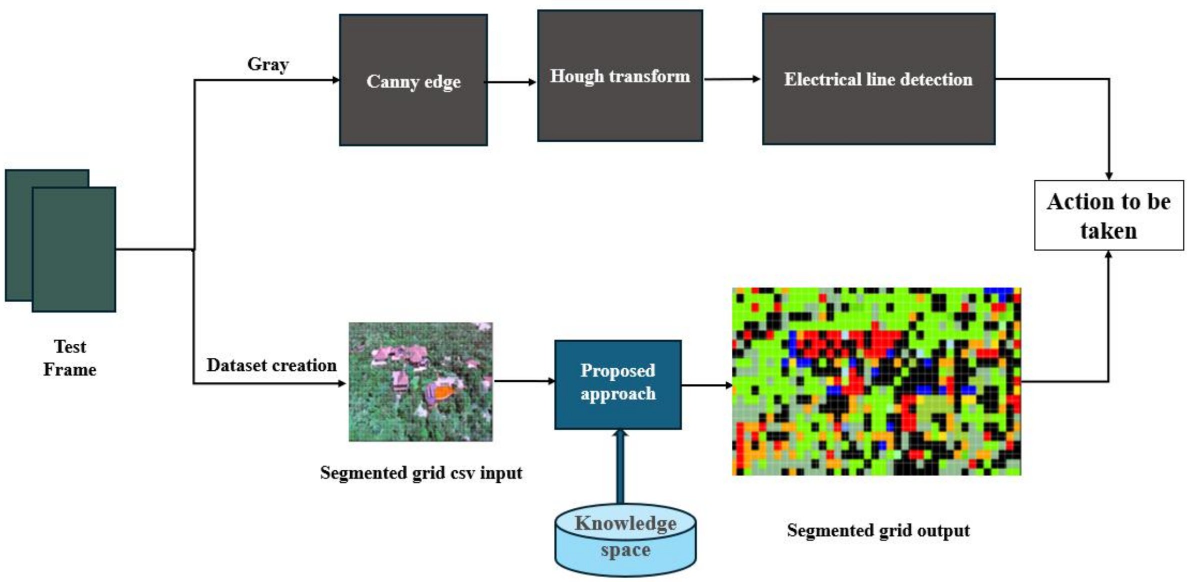


Figure.4.1 Architecture of VCML

Hough Transform: The Hough Transform is a feature extraction technique used in image processing to detect shapes, particularly lines or curves, within an image. It is particularly effective in scenarios where traditional edge detection techniques may fail due to noise or discontinuities in the edge map. The Hough Transform works by mapping points in the image space to lines or curves in a parameter space, where each point in the image corresponds to a sinusoidal curve or line in the parameter space. By accumulating votes along these curves or lines, the Hough Transform identifies the parameters (e.g., slope and intercept for lines) that represent the most prominent features in the image. This technique is widely used in applications such as lane detection in autonomous driving, line detection in document analysis, and circle detection in object recognition.

Canny Edge: Canny Edge Detection is a popular edge detection algorithm widely used in image processing for detecting edges within images while minimizing noise and preserving edge continuity. The algorithm involves several steps, including smoothing the image with a Gaussian filter to reduce noise, computing the gradient magnitude and direction to identify potential edge pixels, performing non-maximum suppression to thin the edges, and finally, applying hysteresis thresholding to determine the final edge pixels. Weak edge pixels are retained only if they are connected to strong edge pixels. Canny Edge Detection is widely used in applications such as object detection, image segmentation, and feature extraction due to its robustness and ability to detect edges accurately.

Electrical Line Detection: This is likely a specific application that detects electrical lines in an image, using techniques such as the Hough Transform and edge detection.

Test Frame: This likely refers to individual frames which is unseen and new.

Dataset Creation: This step involves preparing and organizing a dataset for testing and evaluating the performance of the proposed approach.

Proposed Approach: This is likely a new method or technique that is Random forest alogorithm that uses the above steps to detect and extract colours for unseen data.

Segmented Grid CSV Input: This might refer to the segmented grid input, which is likely a structured format (CSV) containing information about the segmented grid.

Knowledge Space: This could refer to a space or database where knowledge is stored, possibly the results of the feature extraction and detection techniques.

Segmented Grid Output: This is the final output of the system, which contains the segmented grid. This refer to the structured format (CSV) containing information about the segmented grid and displays the model predicted colour in the empty frame of an original image.

Action to be taken: If the electrical lines is detected on top of the vegetation area or is navigating through it then the forest officers will be notified about it

4.2 ARCHITECTURE DIAGRAM

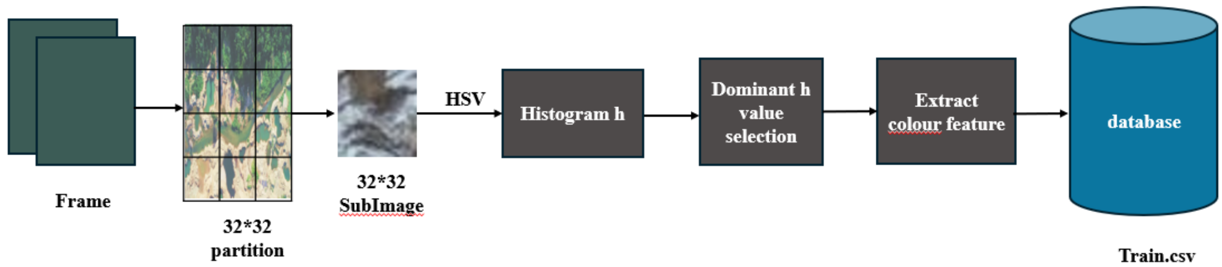


Figure.4.2 Architecture of Dataset Creation

In our approach, we start by dividing the input image into smaller sub-images as shown in Figure 4.3, each measuring 32x32 pixels. These sub-images are then processed in two different ways. Each sub-image goes through these different ways :

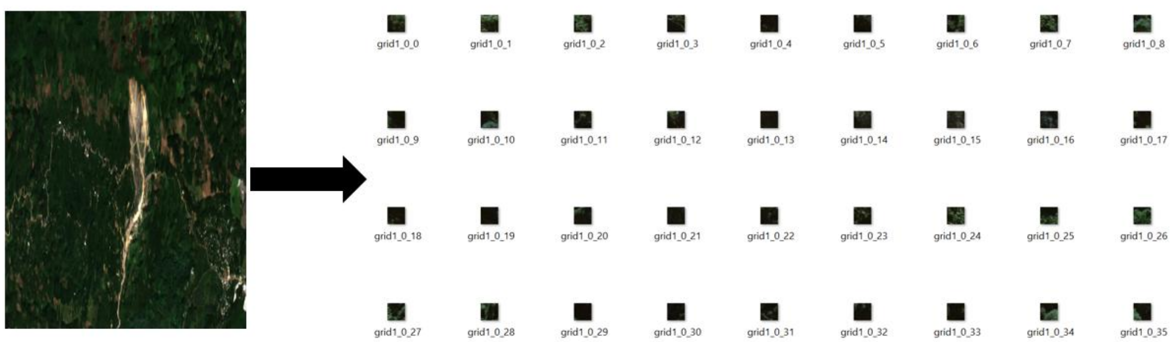


Figure.4.3 Picture Describing Sub-image division visualization.

HSV Image Processing:

Histogram Calculation: We calculate the histogram of the sub-image in the HSV color space, which provides information about the distribution of colors.

Dominant Hue Selection: From the histogram, we identify the dominant hue in the sub-image, which can help in identifying the predominant color.

Color Feature Extraction: Using the dominant hue information, we extract color features from the sub-image and store them in the database for future reference.

By processing the sub-images in these two ways, we aim to capture both structural and color information from the input image, which can be valuable for various applications such as object detection, image retrieval, and scene understanding.

Gray Image Processing:

Canny Edge Detection: We use the Canny edge detection algorithm to find the edges in the sub-image, which helps in identifying the boundaries of objects.

Hough Transform: The Hough transform is applied to detect lines in the sub-image, particularly focusing on electrical lines or structures.

Electrical Line detection: This is used to check if the electrical line passing on top of vegetation area or non-vegetation area.

4.2.1 ARCHITECTURE OF PROPOSED MODEL

In our proposed model, depicted in Figure 4.4, we leverage the robust capabilities of Random Forest. Random Forest is an ensemble learning method renowned for its versatility and effectiveness in both classification and regression tasks. By constructing a multitude of decision trees during training and aggregating their predictions through voting or averaging, Random Forest mitigates overfitting and enhances generalization performance. Its ability to handle high-dimensional data, deal with missing values, and provide feature importance analysis makes it a popular choice across diverse domains. With our implementation, we harness the power of Random Forest to deliver reliable and accurate predictions, essential for addressing the challenges in our target application domain.

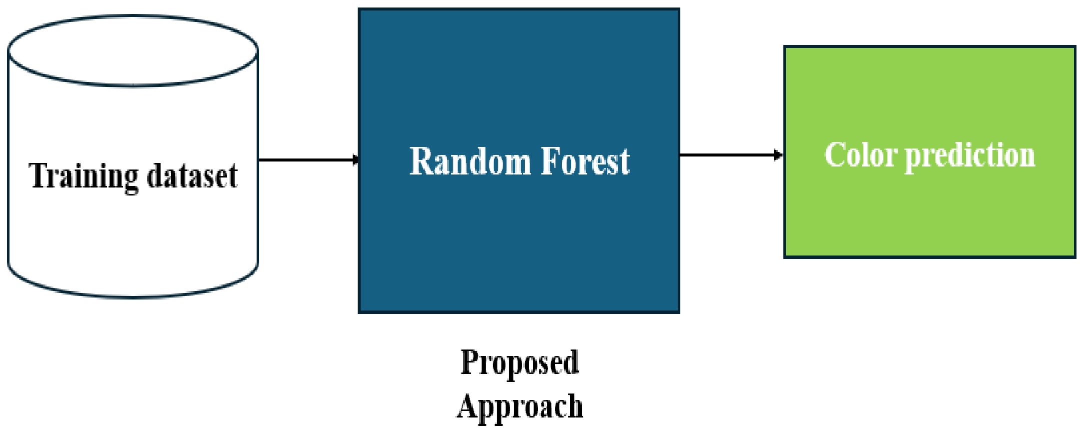


Figure.4.4 Proposed Model

Data Retrieval:

The stored color features are retrieved from the database for each sub-image. These color features may include various metrics such as color histograms, color moments, or color descriptors extracted from the image pixels. Each sub-image represents a portion of the original satellite or aerial image.

Model Training:

Four machine learning models - Random Forest, Gradient Boosting, k-Nearest Neighbors (KNN), and Support Vector Machine (SVM) - are applied to the extracted features. These models are trained using a labeled dataset, where each sample is associated with a class label (e.g., vegetation or non-vegetation). During training, the models learn the patterns and relationships between the extracted features and their corresponding classes, enabling them to classify vegetation cover accurately.

Model Evaluation:

The performance of each model is evaluated using a validation dataset. This dataset contains samples that were not used during the training phase and serves to assess the generalization ability of the models. Performance metrics such as accuracy, precision, recall, and F1-score are calculated to determine the most accurate and dominant model for classifying vegetation cover.

Output Selection:

Based on the evaluation results, the Random Forest model is selected as it provided the highest accuracy in classification. Random Forest is chosen for its ability to handle high-dimensional feature spaces, robustness to overfitting, and ease of interpretation.

Dominant Color Extraction:

For each sub-image, the dominant color is extracted based on the output of the Random Forest model. The model predicts the class label (e.g., vegetation or non-vegetation) for each sub-image, and the dominant color associated with the predicted class is selected.

Final Output:

The dominant colors extracted from each sub-image are placed onto the original image, representing the dominant colors of vegetation cover across the entire scene. This approach provides a visually interpretable representation of vegetation distribution, aiding in environmental monitoring, conservation efforts, and land management practices. The final output serves as a valuable resource for stakeholders to understand vegetation patterns and make informed decisions related to land use and conservation.

4.3 USE CASE DIAGRAM

A use case diagram is a visual representation of the interactions between actors (users or external systems) and the system being modeled. It depicts the functionalities and behavior of the system from an external perspective. Each use case represents a specific functionality or action that the system provides to its users.

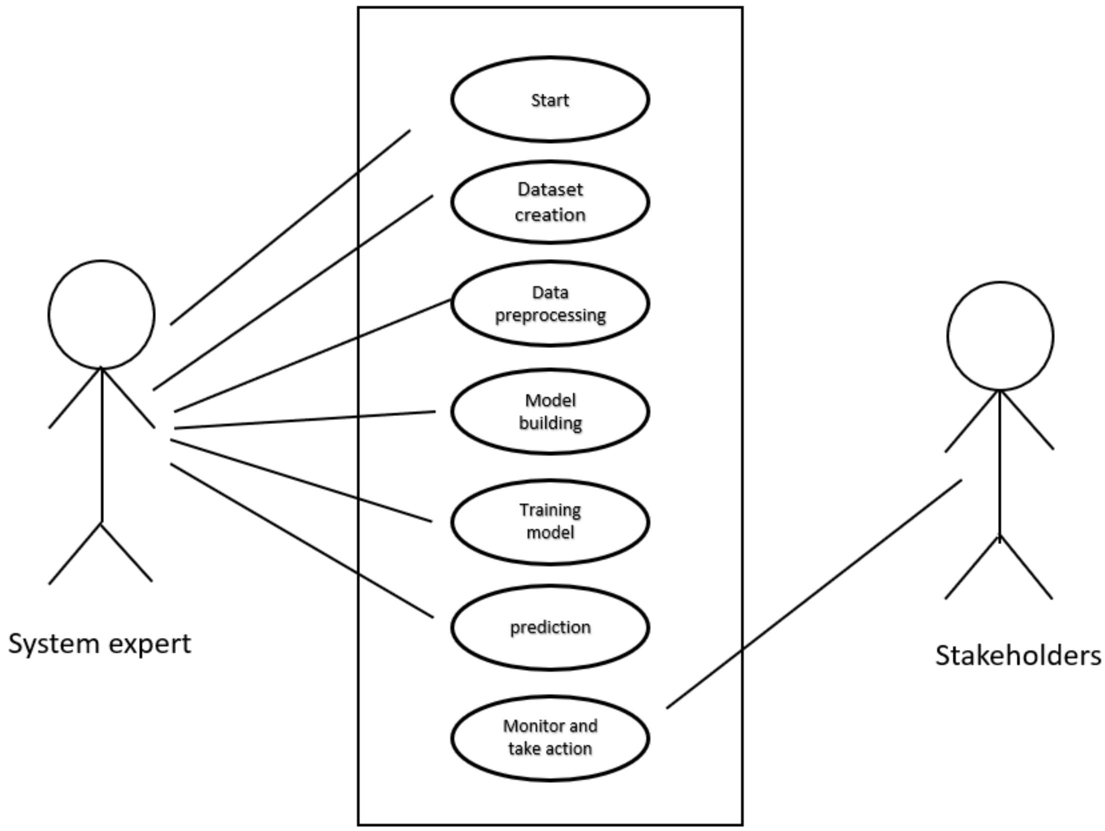


Figure.4.5 Use Case Diagram

The above Figure 4.5 illustrates the interactions between actors (users or external systems) and the system being designed. In the context of vegetation measurement along a line corridor using satellite imagery, the primary actors typically include the system users and any external systems or databases involved. Use cases represent the various functionalities or tasks that users can perform with the system. These functionalities are depicted as ellipses connected by lines to actors, indicating which actors participate in each use case. Use Case Diagrams provide a high-level overview of the system's functionality and help identify the main features and interactions required to achieve the project's objectives.

4.4 ACTIVITY DIAGRAM

An activity diagram is a type of behavior diagram in UML (Unified Modeling Language) that represents the flow of actions or activities within a system or process. It focuses on the sequencing of activities and the relationships among them, providing a visual representation of the dynamic behavior of a system.

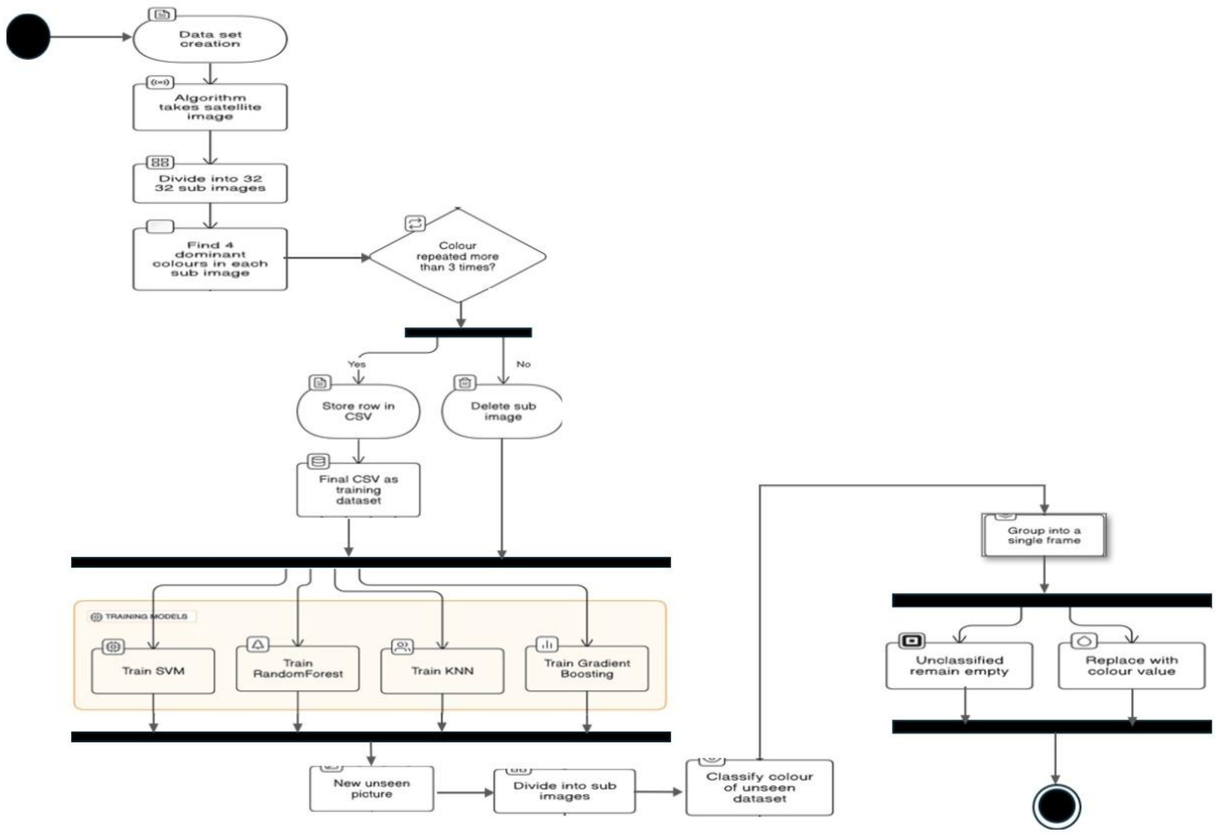


Figure.4.6 Activity Diagram

The above Figure 4.6 depicts the flow of activities or tasks within a system or process. In the context of vegetation measurement along a line corridor using satellite imagery, an Activity Diagram can illustrate the sequence of steps involved in each major process, such as dataset creation, data preprocessing, model building, training model, prediction, and monitoring. Each activity is represented as a rounded rectangle, and arrows indicate the flow of control from one activity to another. Decision points, represented by diamonds, indicate branching or conditional logic within the process. Activity Diagrams help visualize the workflow, identify potential bottlenecks or inefficiencies, and ensure that all necessary steps are included in the process.

4.5 SEQUENCE DIAGRAM

A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together. Here, it portrays how the user, client application, and server interact. This can be seen clearly in the below figure.

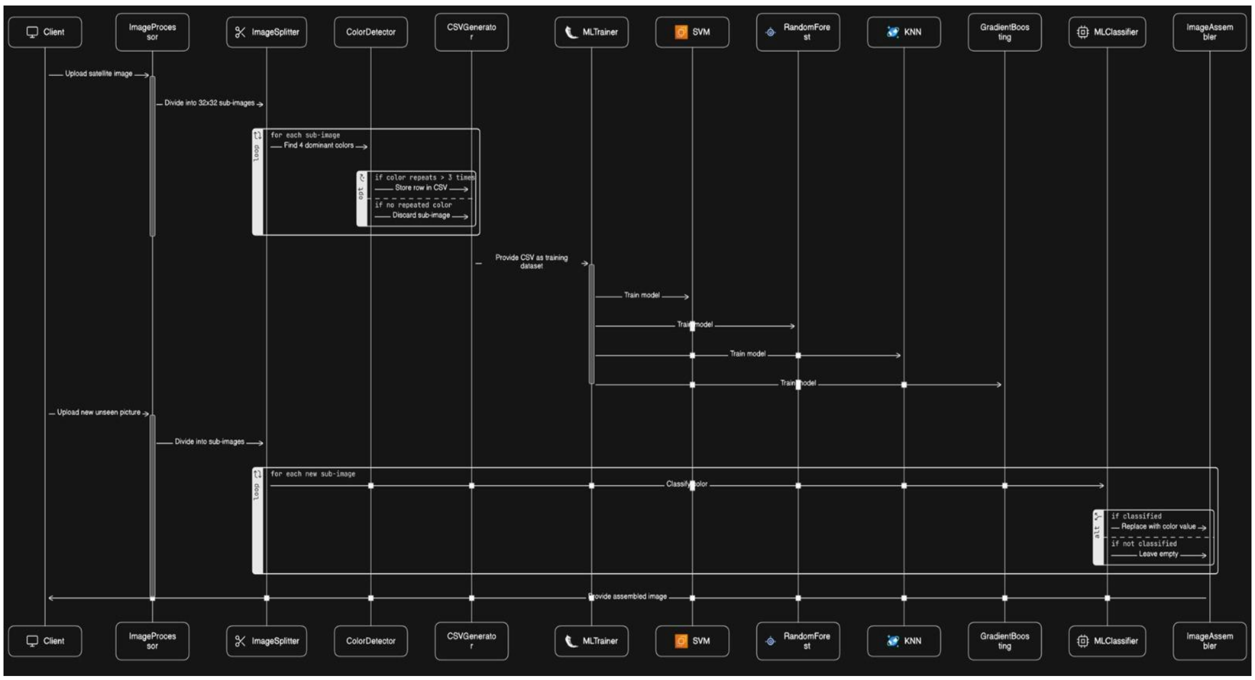


Figure.4.7 Sequence Diagram

The above Figure 4.7 illustrates the interactions between objects or components within a system over time. In the context of vegetation measurement along a line corridor using satellite imagery, a Sequence Diagram can show the sequence of messages exchanged between system components during tasks such as dataset creation, model building, training, prediction, and monitoring. Each object or component involved in the interaction is represented as a vertical lifeline, and messages between objects are depicted as arrows connecting lifelines. Sequence Diagrams help visualize the dynamic behavior of the system, including the order of operations and the flow of data between components. They are particularly useful for understanding the detailed interactions and dependencies between different parts of the system during execution.

CHAPTER 5
IMPLEMENTATION AND TESTING

5.1 IMPLEMENTATION

5.1.1 Images:

Some of the sample images which we took for this implementation are shown in the following Figure 5.1.

Figure.5.1 Sample Data Images

In our project focused on vegetation measurement along a line corridor, we've harnessed the rich data captured by three distinct imaging platforms: drones, aerial surveys, and satellite imagery. By leveraging these diverse sources, we aim to obtain a comprehensive understanding of vegetation dynamics along the corridor. Each imaging platform offers unique advantages: drones provide high-resolution, localized data suitable for detailed analysis; aerial surveys offer broader coverage and context; while satellite imagery ensures global-scale observations. By integrating information from these sources, we can enhance the accuracy and scope of our vegetation measurement efforts, enabling effective environmental monitoring and management.

5.1.2 Train Dataset:

The code snippets comprise several functions tailored for analyzing satellite images and extracting data on color frequencies within 32x32 pixel grids.

Firstly, the get_color_frequency(image) function computes the occurrence of each color in the image by iterating through its pixels, converting RGB values into tuples, and tallying the occurrences of each color, ultimately returning a dictionary containing these frequencies. Following this, the 'rgb_to_hsv(rgb)' function converts RGB color tuples to their corresponding HSV representations, employing the 'colorsys.rgb_to_hsv' method for this conversion, and returns the resultant HSV tuple.

Subsequently, the 'classify_color(rgb)' function classifies RGB colors into predefined categories based on their HSV values, establishing hue-based ranges for different color categories and mapping each color accordingly, eventually returning the name of the color category.

The 'divide_image(image)' function partitions the input image into 32x32 pixel grids, processing each grid individually by extracting it from the image and saving it as a separate PNG file.

Lastly, the 'write_output_to_csv(image, path)' function compiles color information for each grid into a CSV file, segregating the image into grids, determining color frequencies for each grid, and arranging colors by frequency.

For each grid, it generates a CSV row encompassing grid indices, dominant colors along with their frequencies, and respective color names, with empty values filling any missing color slots if fewer than four dominant colors are identified. Collectively, these functions enable the efficient analysis of satellite imagery, extraction of color data, and organization of findings into a CSV format for subsequent analysis or application. Through the above process the dataset is created, the sample is shown in the below Table 5.1.

grid	color1_h	color1_s	color1_v	color2_h	color2_s	color2_v	color3_h	color3_s	color3_v	color4_h	color4_s	color4_v	repeated_colour
Grid(0, 1)	12	26	13	13	27	14	9	23	10	11	25	12	Bright Green
Grid(0, 2)	16	35	15	18	35	16	15	34	15	14	33	13	Lawn Green
Grid(0, 3)	13	30	12	15	32	14	14	31	13	16	38	17	Lawn Green
Grid(0, 4)	9	21	9	10	22	10	8	20	8	18	37	18	Bright Green
Grid(0, 8)	18	37	17	18	36	14	14	35	16	20	44	18	Lawn Green
Grid(0, 9)	11	28	12	14	31	15	12	29	13	20	39	17	Bright Green
Grid(0, 10)	17	42	13	18	43	14	23	48	19	22	47	18	Lawn Green
Grid(0, 11)	17	40	14	22	47	18	15	37	14	16	39	13	Lawn Green
Grid(0, 13)	28	51	23	19	38	16	12	31	12	18	40	17	Lawn Green
Grid(0, 14)	11	28	10	13	27	10	10	27	11	9	26	8	Lawn Green
Grid(0, 15)	20	44	20	9	23	8	8	22	7	7	21	6	Lawn Green
Grid(0, 16)	16	38	17	17	39	18	15	37	16	18	40	19	Bright Green
Grid(0, 18)	17	36	17	20	39	19	18	37	17	19	38	18	Lawn Green
Grid(0, 19)	20	39	19	19	41	18	19	38	18	18	40	17	Lawn Green
Grid(0, 20)	27	31	16	23	30	14	39	44	24	46	51	31	Inchworm
Grid(0, 21)	14	31	15	25	38	20	23	38	19	18	31	14	Lawn Green
Grid(0, 22)	19	38	18	18	37	17	20	39	19	23	42	22	Lawn Green
Grid(0, 23)	20	38	16	19	38	16	20	39	17	21	40	18	Lawn Green
Grid(0, 24)	20	39	20	24	46	25	17	36	16	8	32	8	Bright Green
Grid(0, 26)	35	60	30	29	59	25	54	60	32	31	58	25	Lawn Green
Grid(0, 28)	20	38	16	19	37	15	19	38	16	18	36	14	Lawn Green
Grid(0, 29)	33	60	27	22	36	19	15	32	13	36	48	24	Lawn Green
Grid(0, 30)	33	49	20	36	47	30	31	50	22	34	20	9	Lawn Green
Grid(0, 32)	15	32	13	16	33	14	14	31	12	20	35	14	Lawn Green
Grid(0, 33)	20	37	19	17	34	16	26	52	25	19	36	17	Lawn Green
Grid(0, 34)	17	36	16	14	33	11	16	39	13	45	58	30	Lawn Green
Grid(0, 35)	17	36	17	18	35	17	15	32	14	19	36	18	Lawn Green
Grid(0, 36)	17	36	16	16	35	15	13	32	13	15	34	14	Lawn Green

Table.5.1 Feature dataset

5.1.3 Models

Random Forest:

Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes (classification) or the average prediction (regression) of the individual trees.

In our code implementation, Random Forest is utilized as one of the primary models for vegetation cover classification due to its simplicity, efficiency, and ability to handle noisy and correlated features.

```python
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

# Model training
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)
rf_classifier.fit(X_train, y_train)

# Model evaluation
y_pred = rf_classifier.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.9524986271279516
```

Figure 5.2 Random forest code Snippet

Gradient Boosting (GB):

Gradient Boosting is another ensemble learning technique that builds a series of weak learners (typically decision trees) sequentially, with each tree learning to correct the errors of its predecessor.

Unlike Random Forest, which builds independent trees in parallel, Gradient Boosting optimizes the overall model by minimizing a loss function through gradient descent.

```python
# Model evaluation
y_pred_gb = gradient_classifier.predict(X_test)
accuracy_gb = accuracy_score(y_test, y_pred_gb)
print("Gradient Boosting Classifier Accuracy:", accuracy_gb)
```

```
Gradient Boosting Classifier Accuracy: 0.9390444810543658
```

Figure 5.3 GB code Snippet

k-Nearest Neighbors (kNN):

k-Nearest Neighbors is a simple and intuitive classification algorithm that works by finding the k closest data points in the feature space and assigning the majority class label among them.It is a non-parametric method, meaning it does not make assumptions about the underlying distribution of the data.

```python
# Model evaluation
y_pred_knn = knn_classifier.predict(X_test)
accuracy_knn = accuracy_score(y_test, y_pred_knn)
print("k-Nearest Neighbors (KNN) Accuracy:", accuracy_knn)
```

```
k-Nearest Neighbors (KNN) Accuracy: 0.8522789676002197
```

Figure 5.4 KNN code Snippet

Support Vector Machine (SVM):

Support Vector Machine is a powerful supervised learning algorithm that constructs a hyperplane in a high-dimensional feature space to separate classes with the maximum margins is effective in handling high-dimensional data and can capture complex decision boundaries through kernel tricks, such as polynomial or radial basis function (RBF) kernels.

```
    svm_classifier = SVC()
    svm_classifier.fit(X_train, y_train)

    # Model evaluation
    y_pred_svm = svm_classifier.predict(X_test)
    accuracy_svm = accuracy_score(y_test, y_pred_svm)
    print("Support Vector Machine (SVM) Accuracy:", accuracy_svm)
```

```
Support Vector Machine (SVM) Accuracy: 0.9239428885227897
```

Figure 5.5 SVM code Snippet

From the above models, we see that Random Forest has highest accuracy as shown in Figure 5.2 than the remaining models. So, for the Image Segmentation we have selected the Random Forest. The accuracies or the models are compared using the graph int the below Figure 5.6.

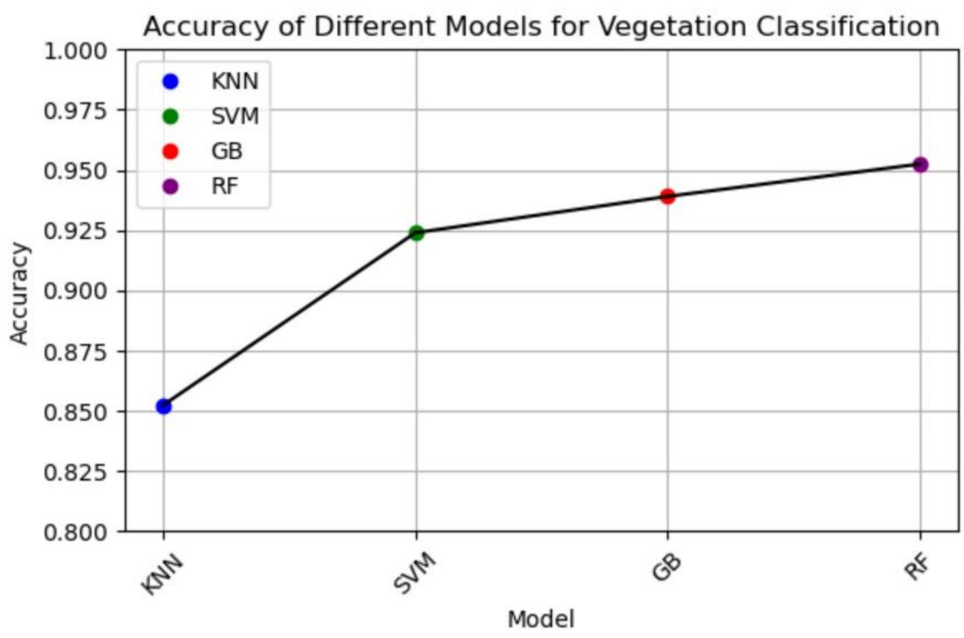


Figure.5.6 Model performance

5.1.4 Image Segmentation

The code snippet commences by converting the predicted color from RGB to HSV format, thus enabling comprehensive analysis grounded in the hue, saturation, and value components. Through meticulous iteration across predefined color ranges, the code meticulously aligns the predicted color with its corresponding HSV value, accounting for nuanced variations inherent within each distinct range. Upon discerning the specific HSV color, a pivotal step ensues, as the code diligently scrutinizes whether it aligns with the verdant spectrum, typically characterized by hue values oscillating between 75 and 165, serving as a telltale signifier of lush vegetation. Drawing upon this critical determination, the code orchestrates a pivotal intervention, deftly assigning a quintessential visual identifier to the corresponding grid region ensconced within the array: a pristine white hue. The below Figure 5.7 demonstrates the code to segment image as mentioned above.

```
hsv_color = None
for color_range in color_ranges:
    if isinstance(color_range[2], list):
        for color_tuple in color_range[2]:
            if color_tuple[0] == predicted_color:
                hsv_color = color_tuple[1]
                break
    else:
        if color_range[2] == predicted_color:
            hsv_color = (color_range[0] + color_range[1]) // 2
            break

# Check if the color falls within the green range
if isinstance(hsv_color, int):
   if 75 <= hsv_color <= 165:
      grid_array[start_x:end_x,start_y:end_y] = (255, 255, 255)  # White for green shades
   else:
      grid_array[start_x:end_x,start_y:end_y] = (0, 0, 0)  # Black for non-green shades
else:
 print("HSV value is not an integer, cannot determine color range.")
```

Figure.5.7 Image Segmentation Code Snippet for black and white

(255, 255, 255) elegantly denotes verdant shades indicative of thriving flora, whereas a stark black hue (0, 0, 0) boldly asserts its presence, marking territories devoid of verdure. This judiciously executed visualization strategy adeptly illuminates the sprawling tapestry of vegetation cover delineated across the expanse of the imagery grid, thereby furnishing invaluable insights into the verdant tableau it seeks to encapsulate.

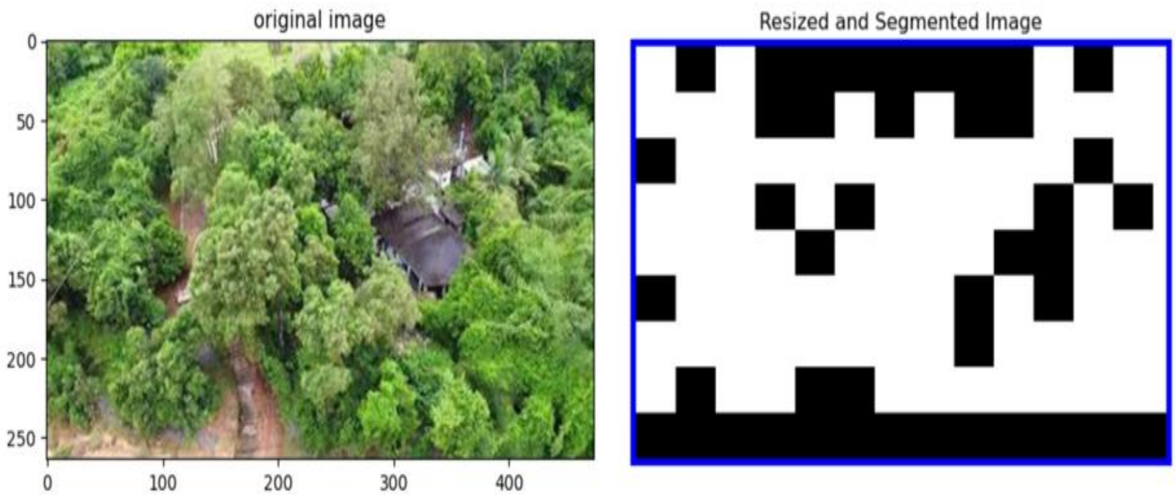


Figure.5.8 Image segmentation (Black and White)

The below code snippet begins by converting the predicted color from RGB to HSV format, allowing for subsequent analysis based on hue, saturation, and value components. It then iterates through predefined color ranges, matching the predicted color with its corresponding HSV value while considering variations within each range. Once the HSV color is identified, the code assigns a corresponding color to the grid region in the array based on the prediction made by the model. These predicted colors are stored in the 'predicted_color' column of the merged_data.csv file. For each grid region, the code retrieves the predicted color from the CSV file and assigns the corresponding RGB value to the grid region within the image array. This process facilitates the visualization of the predicted color distribution within the imagery grid as shown in Figure 5.9, aiding in the analysis of the model's predictions. Additionally, the code draws gridlines on both the original and segmented images to enhance visual clarity. Finally, it saves the segmented image and displays both the original and segmented images for comparison.

```
# Fill the 2D array with color indexes based on the predicted_color column
for index, row in merged_data.iterrows():
    grid_str = row['grid']
    match = re.match(r'Grid\((\d+), (\d+)\)', grid_str)
    if match:
        grid_x = int(match.group(1))
        grid_y = int(match.group(2))
        #print(grid_x,',',grid_y)
        if 0 <= grid_x < 8 and 0 <= grid_y < 14:
            color_name = row['predicted_color']
            cv=color_map[color_name]
            #print(color_name,cv)
            image_array[grid_x*32:grid_x*32+32, grid_y*32:grid_y*32+32,:] = cv
        else:
            print(f"Grid coordinates ({grid_x}, {grid_y}) are outside the bounds of the grid")
    else:
        print(f"Could not extract grid coordinates from '{grid_str}'")

# Function to draw the grid on an image
def draw_grid(image):
    # Calculate the number of grid lines
    num_horizontal_lines = image.shape[0] // grid_spacing
    num_vertical_lines = image.shape[1] // grid_spacing

    # Draw the horizontal and vertical grid lines
    for i in range(num_horizontal_lines):
        start_point = (0, i * grid_spacing)
        end_point = (image.shape[1], i * grid_spacing)
        cv2.line(image, start_point, end_point, (255, 255, 255), 1)

    for i in range(num_vertical_lines):
        start_point = (i * grid_spacing, 0)
        end_point = (i * grid_spacing, image.shape[0])
        cv2.line(image, start_point, end_point, (255, 255, 255), 1)
```

Figure.5.9 Image Segmentation Code Snippet for color

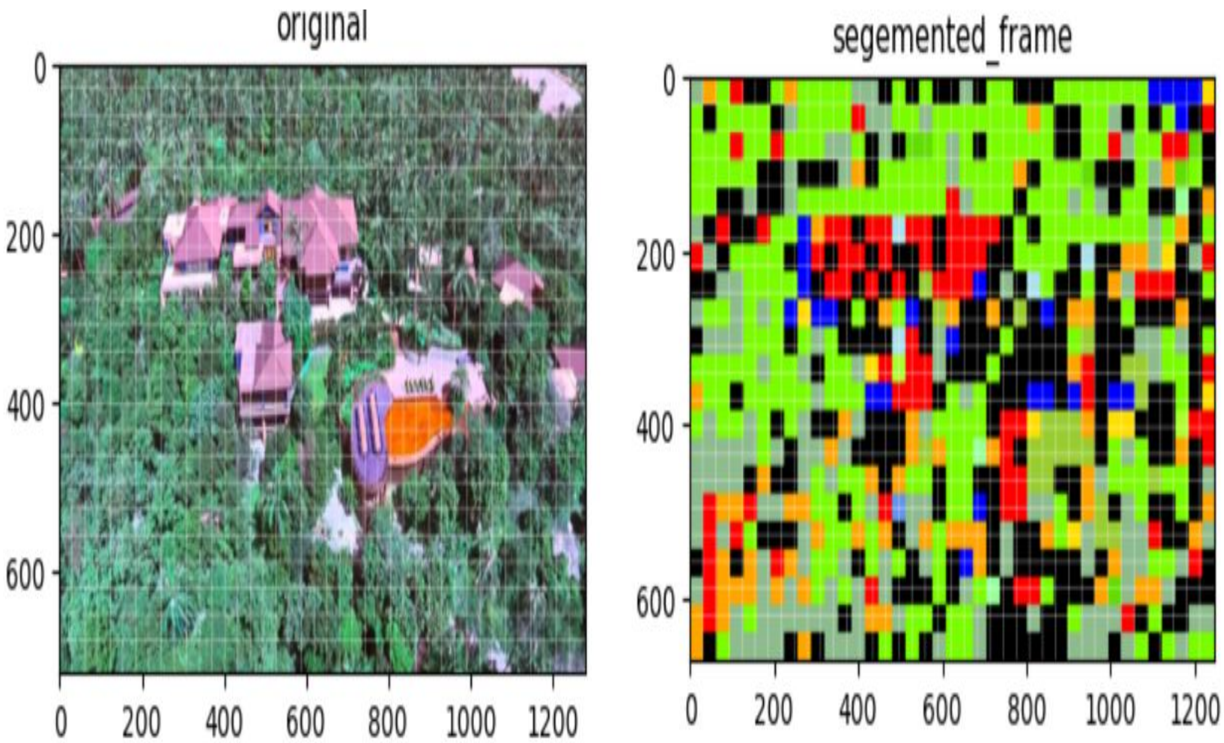


Figure.5.10 Image segmentation (color)

5.1.5 Power Line Detection

Approach 1: Power Lines Detection using Hough Transform

The below code snippet is designed to detect electric lines in an input image using the Hough Line Transform technique. The `houghtransform` function applies the Hough Line Transform on a given image patch to detect lines, while the `process_image` function divides the input image into smaller patches (32x32) and checks each patch for the presence of electric lines using the `houghtransform` function. If electric lines are detected in a patch, it marks the corresponding area in the output image with white color; otherwise, it marks it with black color. The script loads an image, processes it using the defined functions, and displays the original and output images side by side using matplotlib. The Hough Line Transform parameters like rho, theta, threshold, minLineLength, and maxLineGap are tuned empirically to suit the task of electric line detection. This code can be useful in scenarios where automatic detection of electric lines in images is required, such as in the field of infrastructure maintenance or urban planning.

```
def houghtransform(subimg):
    gray = cv2.cvtColor(subimg, cv2.COLOR_BGR2GRAY)
    _, binary = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)
    edges = cv2.Canny(binary, 3, 5, apertureSize=3)
    lines = cv2.HoughLinesP(edges, rho=1, theta=np.pi/180, threshold=20, minLineLength=25, maxLineGap=5)
    if lines is not None:
        return True
    else:
        return False

def process_image(inputframe):
    m, n, r = inputframe.shape
    outputframe = np.zeros(inputframe.shape)
    for x in range(0, m, 32):
        for y in range(0, n, 32):
            subimg = inputframe[x:x+32, y:y+32,:]
            electricline = houghtransform(subimg)
            if electricline:
                outputframe[x:x+32, y:y+32,:] = (255,255,255)
            else:
                outputframe[x:x+32, y:y+32,:] = (0, 0, 0)
    return outputframe
```

Figure.5.11 Code snippet for Power Line detection Approach 1

The code utilizes the Hough Line Transform method to identify the presence of straight lines, particularly electric lines, within an image. By breaking down the image into smaller patches and applying the transform, it determines if any straight lines are present. This approach is valuable for tasks requiring automated detection of such features, such as infrastructure monitoring or urban development planning.

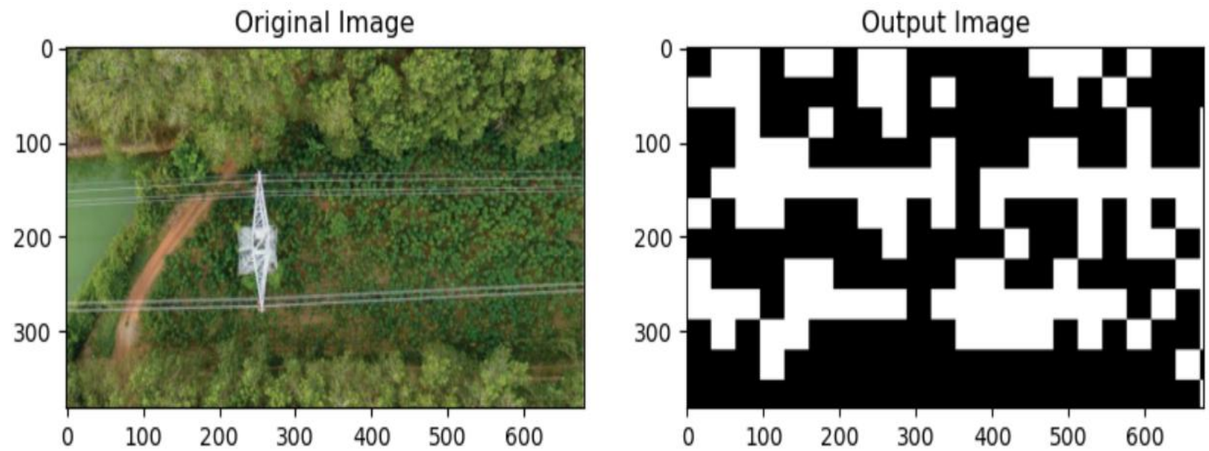


Figure.5.12 Power line detection using Hough transform for 32*32 sub-images.

Approach 2: Power Lines Detection based on Pixel Color Ranges using Hough Transform

This code snippets as shown in Figure 5.13 aims to detect electric lines in an image by analyzing pixel colors within specific ranges and utilizing the Hough Line Transform algorithm. The `houghtransform` function applies edge detection using the Canny algorithm on a given sub-image, then performs the Hough Line Transform to detect lines. The `process_image` function iterates over each pixel in the input image, extracts the RGB values, and checks if they fall within predefined color ranges associated with electric lines. If a pixel's color matches any of these ranges, a 1x1 sub-image containing only that pixel is created and passed to the `houghtransform` function. If electric lines are detected in that sub-image, the corresponding pixel in the output image is set to white; otherwise, it's set to black. The original and output images are displayed side by side using matplotlib for visualization. This approach can be useful in scenarios where electric lines need to be detected in images, such as in infrastructure inspection or maintenance tasks.

```python
def houghtransform(subimg):
    edges = cv2.Canny(subimg, 50, 150, apertureSize=3)
    lines = cv2.HoughLinesP(edges, rho=1, theta=np.pi/180, threshold=20, minLineLength=100, maxLineGap=10)
    if lines is not None:
        return True
    else:
        return False

def process_image(inputframe):
    outputframe = np.zeros(inputframe.shape, dtype=np.uint8)
    for x in range(inputframe.shape[0]):
        for y in range(inputframe.shape[1]):
            pixel = inputframe[x, y]
            r, g, b = pixel
            if (0 <= r <= 192 and 0 <= g <= 192 and 0 <= b <= 192) or (0 <= r <= 120 and 0 <= g <= 120 and 0 <= b <= 120)
                subimg = np.array([[pixel]])
                electricline = houghtransform(subimg)
                if electricline:
                    outputframe[x, y] = (255, 255, 255)  # White for electric lines
                else:
                    outputframe[x, y] = (0, 0, 0)  # Black for no electric lines
            else:
                outputframe[x, y] = pixel  # Copy pixel as it is
    return outputframe
```

Figure.5.13 Code snippet for Power Line detection Approach 2

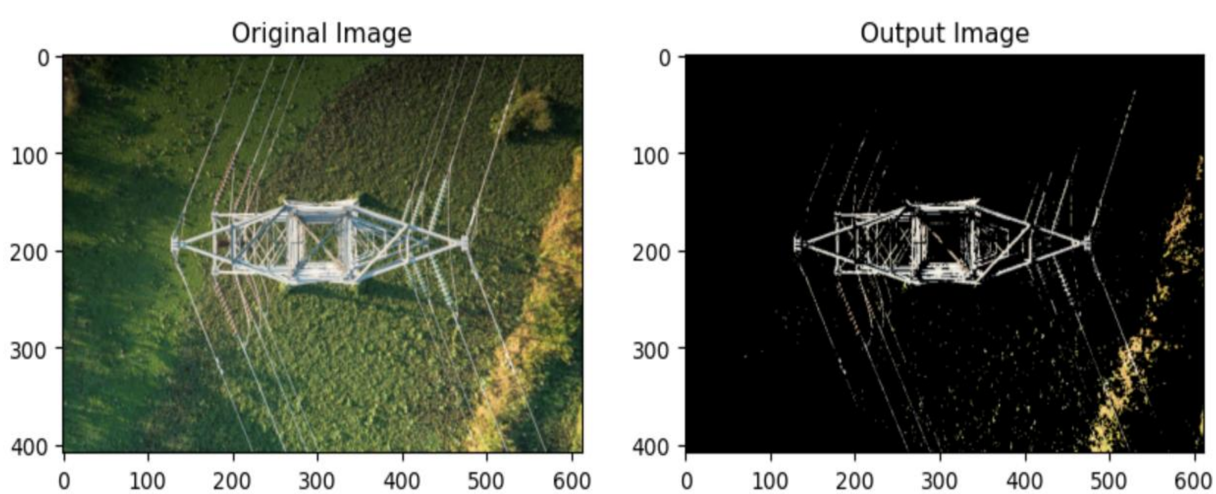


Figure.5.14 Power line detection based on pixel color ranges using Hough transform.

5.2 TESTING

This Python script is designed to evaluate the performance of a classification model using metrics such as classification report, confusion matrix, and various rates including True Positive Rate (TPR), False Positive Rate (FPR), True Negative Rate (TNR), and False Negative Rate (FNR).

First, it loads a CSV file containing the true labels and predicted labels. The true labels represent the original classes of the images, while the predicted labels represent the classes assigned by the segmentation model.

Then, it generates a classification report using the `classification_report` function from scikit-learn, providing metrics such as precision, recall, F1-score, and support for each class.

Next, it computes the confusion matrix using the `confusion_matrix` function from scikit-learn, which tabulates the number of true positives, false positives, true negatives, and false negatives.

After computing the confusion matrix, it calculates the TPR, FPR, TNR, and FNR using the counts from the confusion matrix.

Finally, it prints out the classification report, confusion matrix, as well as the calculated TPR, FPR, TNR, and FNR. These metrics provide insights into the model's performance, indicating its ability to correctly classify instances of different classes and its propensity for making type I and type II errors. This analysis aids in understanding the strengths and weaknesses of the classification model, assisting in further refinement or optimization if necessary.

The below is the sample csv file.

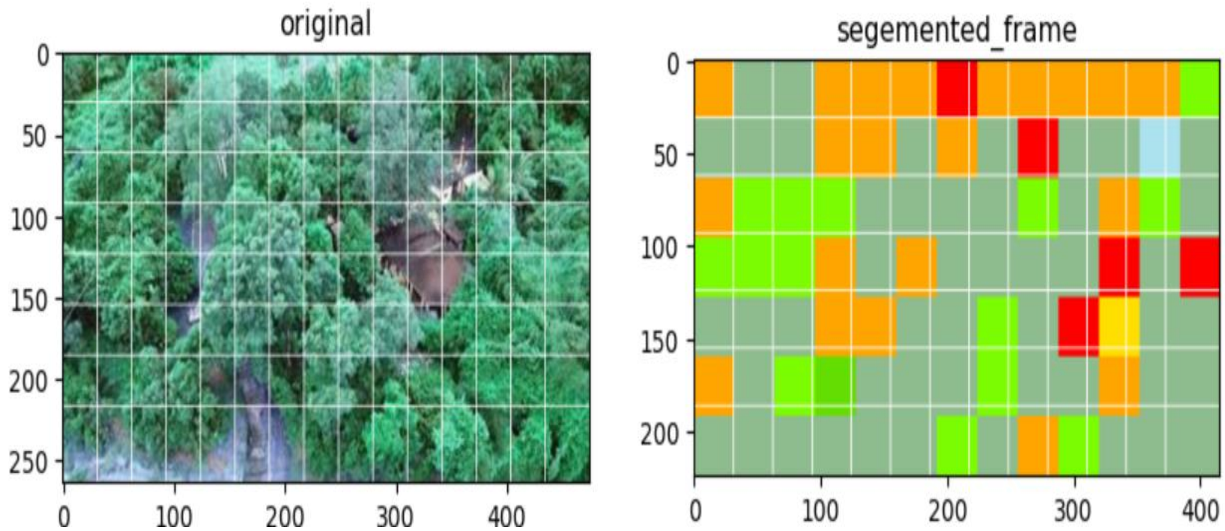


Figure.5.15 Original and Segmented image for classification report

Below is the generated csv by comparing original image and segmented image we have done this process manually and this is defined as ground truth and by considering the below classify.csv we constructed classification report.

block	original_im	segmented_output
1	vegetation	vegetation
2	vegetation	vegetation
3	vegetation	vegetation
4	vegetation	vegetation
5	vegetation	non_vegetation
6	vegetation	non_vegetation
7	vegetation	non_vegetation
8	vegetation	non_vegetation
9	vegetation	non_vegetation
10	vegetation	non_vegetation
11	vegetation	non_vegetation
12	vegetation	non_vegetation
13	vegetation	vegetation
14	vegetation	vegetation

Table.5.2 Classify.csv

The code snippets performs evaluation of a classification model using common metrics such as classification report and confusion matrix, and calculates additional performance measures including True Positive Rate (TPR), False Positive Rate (FPR), True Negative Rate (TNR), and False Negative Rate (FNR).

Firstly, the code loads a CSV file containing the true labels (`original_image`) and predicted labels (`segmented_output`). It then generates a classification report using the `classification_report` function from scikit-learn, providing metrics like precision, recall, F1-score, and support for each class. Following that, a confusion matrix is computed using the `confusion_matrix` function, which tabulates the counts of true positives, false positives, true negatives, and false negatives.

After obtaining the confusion matrix, the code calculates the TPR, FPR, TNR, and FNR using the counts from the confusion matrix. These performance measures are defined as follows:

- True Positive Rate (TPR): The proportion of actual positive cases that were correctly identified by the model. Mathematically is defined in Equation 5.1

TPR = {TP}/ {TP + FN}. (5.1)

- False Positive Rate (FPR): The proportion of actual negative cases that were incorrectly classified as positive by the model. Mathematically is defined in Equation 5.2

FPR ={FP}/ {FP + TN}. (5.2)

- True Negative Rate (TNR): The proportion of actual negative cases that were correctly identified as negative by the model. Mathematically is defined in Equation 5.2

TNR = {TN}/ {FP + TN}.}. (5.3)

- False Negative Rate (FNR): The proportion of actual positive cases that were incorrectly classified as negative by the model. Mathematically is defined in Equation 5.2

FNR = {FN}/ {TP + FN}.}. (5.4)

These metrics provide insights into the model's performance across different classes and can aid in assessing its strengths and weaknesses as shown in Figure 5.16. They are particularly useful in binary classification tasks but can be adapted for multiclass scenarios as well.

```python
# Generate confusion matrix
cm = confusion_matrix(true_labels, predicted_labels)
print("\nConfusion Matrix:")
print(cm)

# Calculate true positive (TP), false positive (FP), true negative (TN), and false negative (FN)
TP = cm[1][1]
FP = cm[0][1]
TN = cm[0][0]
FN = cm[1][0]

# Calculate true positive rate (TPR), false positive rate (FPR), true negative rate (TNR), and false negative rate (FNR)
TPR = TP / (TP + FN)
FPR = FP / (FP + TN)
TNR = TN / (FP + TN)
FNR = FN / (TP + FN)

# Print TPR, FPR, TNR, and FNR
print("\nTrue Positive Rate (TPR):", TPR)
print("False Positive Rate (FPR):", FPR)
print("True Negative Rate (TNR):", TNR)
print("False Negative Rate (FNR):", FNR)
```

Figure.5.16 Classification Report For a sample Image code snippet.

Below is the output for the classification report for the generated classify.csv file. which we have done for three images manually that is comparing original image with the segmented _image output.

Image	Precision	Recall	F1-Score	Accuracy	True Positive Rate (TPR)	False Positive Rate (FPR)	True Negative Rate (TNR)	False Negative Rate (FNR)
Image 1	0.88	0.70	0.76	0.70	0.7037	0.3333	0.6667	0.2963
Image 2	0.82	0.65	0.70	0.65	0.65	0.375	0.625	0.35
Image 3	0.85	0.68	0.74	0.68	0.6818	0.3	0.7	0.3182
Average	0.85	0.68	0.73	0.68	0.6785	0.3361	0.6642	0.3215

Table.5.3 Test Analysis

The above Table 5.3 presents the classification performance metrics for three distinct images utilized in a vegetation measurement project. Each image underwent classification analysis, resulting in metrics such as precision, recall, F1-score, accuracy, true positive rate (TPR), false positive rate (FPR), true negative rate (TNR), and false negative rate (FNR). These metrics offer insights into the effectiveness of the classification models applied to each image dataset. By comparing the average metrics across the three images, a holistic assessment of the classification performance can be made, highlighting trends and variations. Additionally, individual analysis of each image provides specific context regarding its classification accuracy and reliability. This comprehensive evaluation aids in understanding the strengths and limitations of the classification approach and guides further refinement or optimization efforts to enhance the project's overall effectiveness in vegetation measurement along line corridor using satellite imagery.

CHAPTER 6
CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

In our project titled "Vegetation Measurement Along Line Corridor Using Satellite Imagery," we employed advanced image processing techniques and machine learning models to accurately assess vegetation cover along a designated corridor. Leveraging satellite imagery as our primary data source, we implemented four distinct classification models, including Support Vector Machine (SVM), k-Nearest Neighbors (KNN), Random Forest, and Gradient Boosting. Through rigorous evaluation, it was determined that the Random Forest model exhibited the highest accuracy of 95.24 among the tested models.

The classification reports generated for each model provided valuable insights into the performance metrics, including precision, recall, F1-score, and accuracy, as well as additional rates such as true positive rate (TPR) and false positive rate (FPR). Analysis of these metrics facilitated a comprehensive assessment of the classification models' effectiveness in distinguishing vegetation from non-vegetation areas within the corridor imagery.

Overall, our project demonstrates the efficacy of machine learning techniques in vegetation measurement tasks, particularly when applied to satellite imagery datasets. By accurately classifying vegetation cover along the corridor, our approach contributes to environmental monitoring efforts and informs land management decisions, ultimately promoting sustainable practices and conservation initiatives. Moving forward, continued refinement and optimization of classification models could further enhance the accuracy and applicability of our vegetation measurement system.

6.2 FUTURE SCOPE

I. **Integration of Deep Learning Layered Models:**

   Incorporate more advanced deep learning architectures, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), to improve the accuracy and robustness of vegetation classification. These models can automatically learn hierarchical features from the satellite imagery, potentially capturing more intricate patterns and relationships within the data.

II. **Exploration of Wavelet Transformation and Fourier Transformation:**

Investigate the application of wavelet transformation and Fourier transformation techniques to extract spatial and spectral features from the satellite imagery. These transformations can help in capturing both local and global variations in vegetation cover, enhancing the discriminative power of the classification models.

III. **Temporal Analysis for Change Detection:**

Extend the project's capabilities to perform temporal analysis by incorporating time-series satellite imagery. By detecting changes in vegetation cover over time, the system can provide insights into vegetation dynamics, seasonal variations, and long-term trends, enabling more comprehensive monitoring and management of the corridor ecosystem.

IV. **Multi-Sensor Fusion:**

Explore the fusion of data from multiple remote sensing platforms, including satellite imagery, aerial surveys, LiDAR data, and ground-based sensors, enabling a more holistic understanding of the corridor environment.

V. **Spatially Explicit Modeling:**

Develop spatially explicit models to account for the heterogeneity of vegetation cover within the corridor. By considering spatial relationships and dependencies, these models can provide more accurate predictions of vegetation distribution and composition, facilitating targeted conservation efforts and land management strategies.

VI. **Interactive Visualization and Decision Support:**

Build interactive visualization tools and decision support systems that allow stakeholders to explore and analyze vegetation data interactively. These tools can provide intuitive interfaces for visualizing vegetation patterns, conducting scenario analysis.

VII. **Integration with Geographic Information Systems (GIS):**

Integrate the project with GIS platforms to leverage spatial analysis capabilities and incorporate additional geospatial datasets (e.g., land use/land cover, topography, hydrology). This integration enables more comprehensive spatial analysis and facilitates the integration of vegetation measurement data into broader environmental management workflows.

REFERENCES

Articles:

[1]. Viereck, Leslie A. The Alaska vegetation classification. Vol. 286. US Department of Agriculture, Forest Service, Pacific Northwest Research Station, 1992.

[2]. Tansley, Arthur G. "The classification of vegetation and the concept of development." The Journal of ecology (1920): 118-149.

[3]. Hirata, Masahiro, Naoki Kogab, Hitoshi Shinjo, Haruhiro Fujita, Gustave Gintzburger, and Akira Miyazaki. "Vegetation classification by satellite image processing in a dry area of north-eastern Syria." International Journal of Remote Sensing 22, no. 4 (2001): 507-516.

[4]. Zhao, Fuchao, Xiaoming Wu, and Shuai Wang. "Object-oriented vegetation classification method based on UAV and satellite image fusion." Procedia Computer Science 174 (2020): 609-615.

[5]. Hashim, Haslina, Zulkiflee Abd Latif, and Nor Aizam Adnan. "Urban vegetation classification with NDVI threshold value method with very high resolution (VHR) Pleiades imagery." The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 42 (2019): 237-240.

[6]. Kovásznay, Leslie SG, and Horace M. Joseph. "Image processing." Proceedings of the IRE 43, no. 5 (1955): 560-570.

Books:

[7]. Rafael C.Gonzalez, Richard Eugene Woods. "Digital image processing", 978-0-13-168728-8, (2006)

[8]. Arcangelo Distante, Cosimo Distante. "Image processing and data vision", 2020

Websites:

[9]. https://www.earthdata.nasa.gov/eosdis.

[10]. https://earth.esa.int/eogateway.

RESEARCH PAPER

Vegetation Measurement along Line Corridor using Satellite Imagery

M. Rishitha Reddy (20B81A0592)
K. Shriya Reddy (20B81A05A3)
S. Pallavi (20B81A0585)
Project Guide: Dr. R. K. SelvaKumar

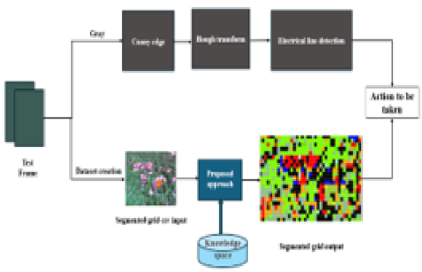
ABSTRACT

Our project utilizes advanced image classification algorithms to monitor vegetation and infrastructure along linear corridors using satellite imagery, addressing challenges in environmental management and infrastructure development. Traditional ground survey methods are time-consuming, costly, and limited in spatial coverage, motivating our use of machine learning and image processing techniques. Our methodology integrates pre-processing, supervised classification for vegetation delineation, and advanced edge detection for identifying electrical lines accurately. Rigorous testing demonstrates a notable 95.24% accuracy in vegetation classification, offering insights for environmental assessments, infrastructure maintenance, and risk mitigation. Our research provides a practical framework for stakeholders in ecosystem monitoring, biodiversity conservation, and land management, enabling informed decision-making in linear corridor projects.

OBJECTIVES

- Develop and deploy advanced image classification algorithms for monitoring vegetation cover using satellite imagery and aerial data, focusing on distinguishing forested areas, agricultural lands, and power lines.
- Create a tool for rapid threat detection like forest fires and illegal deforestation, aiding habitat preservation and power line monitoring.
- Integrate machine learning algorithms to enhance monitoring, detecting changes in vegetation cover, especially distinguishing between forested areas, agricultural lands, and power line corridors, utilizing multi-spectral and high-resolution imagery.

METHODOLOGY



Our methodology is an integrated approach combines color analysis, machine learning, and line detection for comprehensive vegetation measurement along line corridors using satellite imagery. The process begins by extracting color frequencies within 32x32 pixel grids from satellite images, followed by vegetation cover classification using Random Forest, achieving the highest accuracy of 95.42% among SVM, KNN, and Gradient Boosting. RGB to HSV conversion facilitates identification of vegetation-rich areas by specific hue ranges. Additionally, power lines detection is enhanced using the Hough Line Transform algorithm for efficient identification within the imagery.

CONCLUSION

1. Applied advanced image processing and machine learning to assess vegetation cover along a specified corridor using satellite imagery, incorporating line corridor detection via Hough Transform.
2. Utilized four classification models (SVM, KNN, Random Forest, and Gradient Boosting), with Random Forest achieving the highest accuracy at 95.24%.
3. Detailed classification reports yielded insights into performance metrics, aiding the comprehensive evaluation of model effectiveness in discerning vegetation from non-vegetation areas within corridor imagery.

Vegetation Measurement along Line Corridor using Satellite Imagery

M. Rishitha Reddy (20B81A0592)
K. Shriya Reddy (20B81A05A3)
S. Pallavi (20B81A0585)
Project Guide: Dr. R. K. SelvaKumar

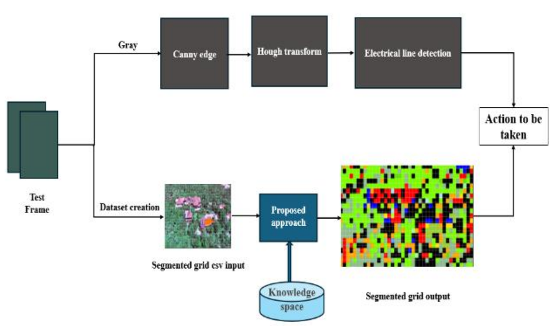
ABSTRACT

Our project utilizes advanced image classification algorithms to monitor vegetation and infrastructure along linear corridors using satellite imagery, addressing challenges in environmental management and infrastructure development. Traditional ground survey methods are time-consuming, costly, and limited in spatial coverage, motivating our use of machine learning and image processing techniques. Our methodology integrates pre-processing, supervised classification for vegetation delineation, and advanced edge detection for identifying electrical lines accurately. Rigorous testing demonstrates a notable 95.24% accuracy in vegetation classification, offering insights for environmental assessments, infrastructure maintenance, and risk mitigation. Our research provides a practical framework for stakeholders in ecosystem monitoring, biodiversity conservation, and land management, enabling informed decision-making in linear corridor projects.

OBJECTIVES

- Develop and deploy advanced image classification algorithms for monitoring vegetation cover using satellite imagery and aerial data, focusing on distinguishing forested areas, agricultural lands, and power lines.
- Create a tool for rapid threat detection like forest fires and illegal deforestation, aiding habitat preservation and power line monitoring.
- Integrate machine learning algorithms to enhance monitoring, detecting changes in vegetation cover, especially distinguishing between forested areas, agricultural lands, and power line corridors, utilizing multi-spectral and high-resolution imagery.

METHODOLOGY



Our methodology is an integrated approach combines color analysis, machine learning, and line detection for comprehensive vegetation measurement along line corridors using satellite imagery. The process begins by extracting color frequencies within 32x32 pixel grids from satellite images, followed by vegetation cover classification using Random Forest, achieving the highest accuracy of 95.42% among SVM, KNN, and Gradient Boosting. RGB to HSV conversion facilitates identification of vegetation-rich areas by specific hue ranges. Additionally, power lines detection is enhanced using the Hough Line Transform algorithm for efficient identification within the imagery.

CONCLUSION

1. Applied advanced image processing and machine learning to assess vegetation cover along a specified corridor using satellite imagery, incorporating line corridor detection via Hough Transform.
2. Utilized four classification models (SVM, KNN, Random Forest, and Gradient Boosting), with Random Forest achieving the highest accuracy at 95.24%.
3. Detailed classification reports yielded insights into performance metrics, aiding the comprehensive evaluation of model effectiveness in discerning vegetation from non-vegetation areas within corridor imagery.