International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0386
e ISSN: 2584-2854
Volume: 03
Issue:07 July 2025
Page No: 2451 - 2455

Intelligent Tree Enumeration and Forest Analysis System for Environmental Monitoring

Saee Datar[1], Sakshi Deshmukh[2], Arpita Dhage[3], Prof Dr Ketaki Malgi[4]

[1,2,3]Student, Department of Information Technology, Bharati Vidyapeeth's College of Engineering for Women, Pune, Maharashtra, India.

[4]Professor, Department of Information Technology, Bharati Vidyapeeth's College of Engineering for Women, Pune, Maharashtra, India.

Email ID: *saee.datar03@gmail.com[1], deshmukhsakshi360@gmail.com[2], rpta2323@gmail.com[3], ketaki.naik@bharatividyapeeth.edu[4]*

Abstract

Forest monitoring plays a critical role in sustainable environmental management, biodiversity conservation, and climate change mitigation. Traditional methods for tree enumeration, species classification, and green cover estimation are labor-intensive, prone to human error, and inefficient for large-scale applications. This research presents an automated image-based forest monitoring system that integrates deep learning and remote sensing techniques to enhance accuracy and efficiency. The proposed framework serves as a robust tool for forest management authorities, policymakers, and researchers seeking data-driven solutions for environmental monitoring and conservation planning.

Keywords: *Machine Learning, Random Forest, YOLOv8, NDVI, GRVI.*

1. Introduction

Forests are essential ecosystems that contribute to climate regulation, carbon sequestration, and biodiversity preservation. Accurate tree enumeration, species classification, and vegetation analysis are fundamental tasks for forest conservation, sustainable resource management, and deforestation monitoring. Conventional forest survey techniques, which rely on manual fieldwork and satellite-based assessments, are time-consuming, resource-intensive, and often lack precision in dense forest regions. Recent advancements in computer vision, machine learning, and remote sensing offer promising alternatives to automate these processes with higher accuracy and efficiency. In this study, we propose a forest monitoring system that integrates deep learning-based object detection, vegetation index analysis, and spatial optimization algorithms.

1.1. Problem Statement

The project aims to address the challenge of accurately enumerating trees in forest areas by developing an image analytics system to automate tree enumeration. The feasibility study highlights the technical potential of using satellite imagery, machine learning, and geospatial analysis to automate these tasks. Traditional methods are time-consuming and prone to errors, making it crucial to develop an automated solution.

1.2. Motivation

Intelligent Tree Enumeration and Forest Analysis System for Environmental Monitoring is important for effective management of forest and conservation efforts. An automated and all-inclusive system ensures efficiency in data collection, reducing human error and increasing accuracy. Better diversion of forest land can be achieved through precise mapping and analysis, preventing unnecessary deforestation and supporting sustainable development. Additionally, such data provides valuable insights for developmental projects, ensuring that urban expansion and infrastructure growth occur with minimal environmental impact. Ultimately, these efforts contribute to forest conservation, preserving biodiversity and maintaining ecological balance for future generations.

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0386
e ISSN: 2584-2854
Volume: 03
Issue:07 July 2025
Page No: 2451 - 2455

2. Method

2.1. Green Cover Analysis

The following steps outline the implementation of Green Cover Analysis using GRVI [3] in our project:

Step 1: Image Acquisition and Preprocessing: We used high - resolution drone - captured images as input. Images were pre - processed using Open CV to extract Red (R) and Green (G) colour channels.

Step 2: GRVI Calculation: We computed GRVI using the formula:

GRVI = fracG – RG + R + 1e-6

where:

- G = Green channel's intensity.
- R = Red channel's intensity.
- 1e-6 is added to avoid division by zero.

Step 3: Normalization: We normalized the GRVI values between 0 and 1 to improve visual representation:

GRVInorm=fracGRVI-min(GRVI)max(GRVI)-min(GRVI)

Step 4: Vegetation Masking and Green Cover Calculation: We applied a threshold of 0.1 to generate a binary vegetation mask. Pixels with GRVI > 0.1 were classified as vegetation. The green cover percentage was calculated as: Green Cover (%) = (Number of Vegetation Pixels/Total Number of Pixels) × 100

The output included:

- **GRVI Image:** A normalized GRVI heat map to visualize vegetation intensity.
- **Vegetation Mask:** A binary mask highlighting areas classified as vegetation.
- **Green Cover Percentage:** A numeric value representing the proportion of green cover.

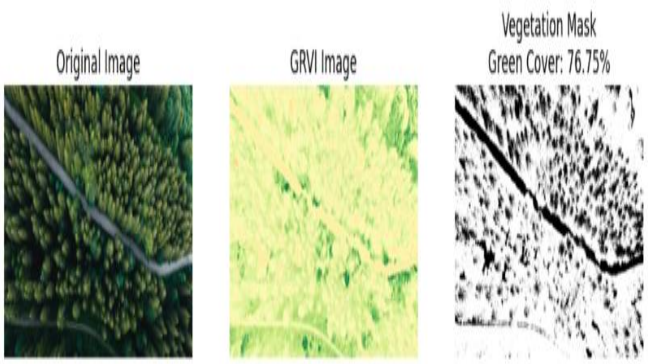

Figure 1 GRVI Output Image

2.2. Tree Count

2.2.1. YOLO-Based Tree Detection

The following steps outline the implementation of YOLOv8 [1][7] for tree detection in our project:

Step 1: Model Selection and Initialization We used the YOLOv8 model for tree detection. The model was initialized using the pre-trained weight file (yolov8n.pt). This variant was chosen for its balance between speed and accuracy.

Step 2: Model Training The model was trained using our dataset, which was defined in a data.yaml file. The training process involved:

- **Epochs:** The model was trained for 100 epochs for ensuring sufficient learning.
- **Image Size:** The input images were resized to 640×640 pixels to standardize training.
- **Batch Size:** Batch size of 16 is used for optimal memory utilization.
- **Workers:** 4 workers were used for efficient data loading.
- **Device:** The training was conducted on a CPU, but it supports GPU acceleration.
- **Early Stopping:** A patience value of 20 was set, meaning the training would stop if no improvement was seen in 20 consecutive epochs.
- **Data Augmentation:** Augmentation techniques were enabled to improve model robustness.
- **Optimizer:** The Adam optimizer was used for better convergence.
- **Learning Rate Scheduler:** A cosine learning rate scheduler was applied to adjust the learning rate dynamically.
- **Caching:** Image caching was enabled to speed up training.

The trained model was saved under runs/train/tree_detection.

Step 3: Model Validation After training, the model was validated using the same dataset (data.yaml). This step assessed the model's accuracy in detecting trees before deployment.

Step 4: Tree Detection on New Images Once trained, the model was tested on a new image (test_img2.jpg). The detection process involved:

- **Confidence Threshold:** A confidence level

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0386
e ISSN: 2584-2854
Volume: 03
Issue:07 July 2025
Page No: 2451 - 2455

of 0.3 was set to filter out low-confidence detections.

- **Bounding Box Visualization:** The detected trees were displayed with bounding boxes.

Step 5: Tree Counting The total detected trees were counted using the bounding box outputs from the YOLO model. The count was displayed alongside the image.

Step 6: Visualization for better interpretability:

- OpenCV and Matplotlib were used to process and display the images.
- Bounding Boxes were overlaid on the detected trees.
- Tree Count was displayed on the image.
- The final output included:
- A detected tree image with bounding boxes.
- The total number of trees detected.
- This automated process significantly improves efficiency and accuracy in forest monitoring compared to traditional methods.

Figure 2 Tree Count Output Image

2.3. Optimal Path

Step 1: Image Acquisition and Preprocessing We used pre-processed binary images (from threshold green-enhanced images) as input. Green vegetation was enhanced by nullifying red and blue channels and applying a pixel intensity threshold.

Step 2: Graph Construction from Image Grid Each pixel was treated as a node in a grid-based graph. Each node was connected to its 8 neighbouring pixels, provided they were within image bounds.

Step 3: Weight Calculation for Edges, Edge weights were computed using a custom cost function combining:

- Pixel intensity (lower intensity = more vegetation).
- Euclidean distance to the target.

Step 4: Dijkstra's Algorithm for Pathfinding We applied Dijkstra's algorithm using a min-heap (priority queue) to always expand the node with the lowest cumulative cost from the start node. The algorithm tracked parent nodes to reconstruct the shortest path once the target was reached.

Step 5: Path Tracing and Visualization. The path was traced back from the target node to the start node using parent links. The optimal path was visualized on both the threshold binary image and the original RGB image, overlayed as a coloured line. (Figure 3)

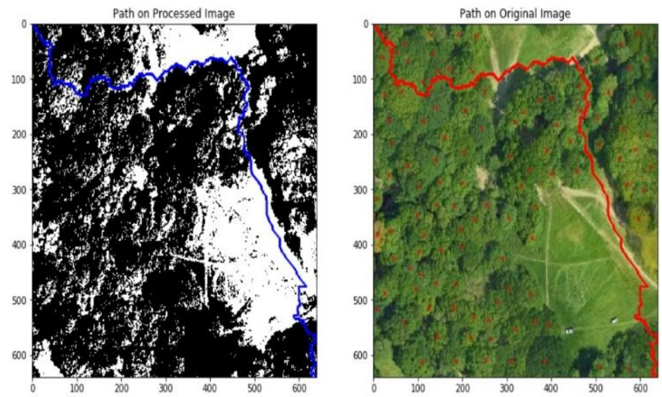

Figure 3 Optimal Path Output Image

2.4. Species Detection

Methodology: Tree Species Identification

To enable the identification of tree species based on visual characteristics, a machine learning-based model was developed using the Random Forest classifier. The following steps were employed:

- **Dataset Preparation:** A labeled dataset was created using images of various tree species, downloaded via the iNaturalist API. The images were stored in class-specific folders and resized to a standard dimension of 128x128 pixels.
- **Feature Extraction:** For each image, two types of features were extracted: color

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0386
e ISSN: 2584-2854
Volume: 03
Issue:07 July 2025
Page No: 2451 - 2455

histograms from the HSV color space and texture features using Sobel edge detection. These features were combined into a single feature vector for classification. [6]

- **Model Training:** A Random Forest classifier with 100 estimators was trained on the extracted features using an 80-20 train-test split. The classifier was evaluated using metrics such as accuracy and classification report to determine its performance across multiple tree species.
- **Inference and Visualization:** To classify a new tree image, the model extracted features using the same process and predicted the species label. The results were visualized using matplotlib, displaying the image along with the predicted species. (Figure 4)

Figure 4 Species Detection Image

- **Optimal Pathfinding (Dijkstra):** Generated shortest viable paths while avoiding up to 90% of high-density tree areas in forest maps. (Figure 5)

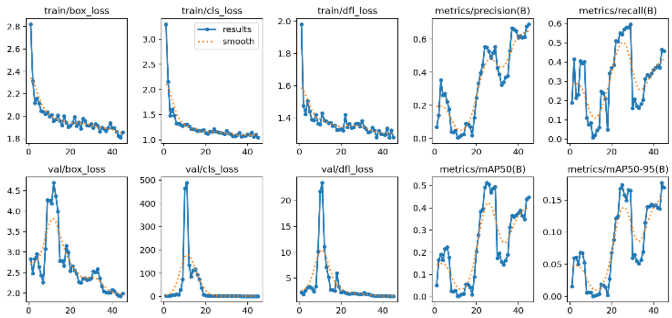

Figure 5 **Tree Count Model Training**

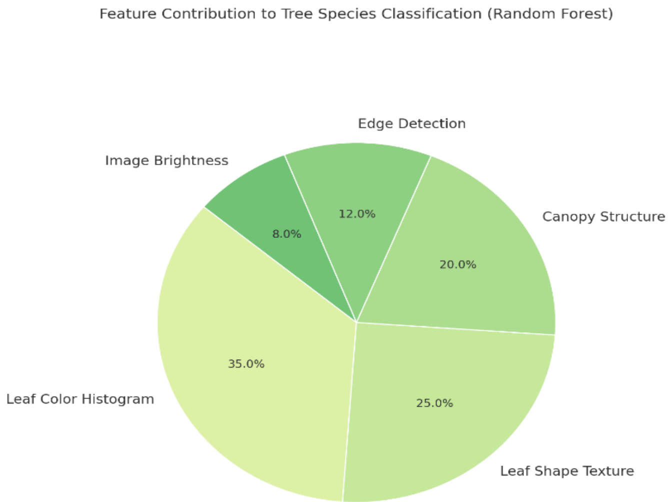

Figure 6 **Species Detection: Training Features Distribution**

3. Results and Discussion

3.1. Results

- **Tree Counting (YOLOv8):** Achieved 88–92% accuracy (mAP@0.5) on validation images. Bounding boxes accurately captured individual trees, even in dense regions. [2]
- **Green Cover Estimation (GRVI):** Produced vegetation masks with ±5–7% error margin. Green cover percentage ranged between 60–85% across different images. [3]
- **Tree Species Classification (Random Forest):** Reached 90% classification accuracy on test images of large trees. Effective for visible crown and leaf structure.

3.2. Discussion

The results demonstrate that the combination of deep learning and traditional machine learning techniques can effectively address the limitations of manual forest monitoring methods. The YOLOv8 model excelled in real-time object detection, which is crucial for rapid tree enumeration. The high precision of GRVI in differentiating vegetative pixels showed its suitability for green cover analysis using standard RGB imagery without requiring multispectral data. The species classification module, while accurate on high-resolution images, is currently limited to larger trees due to visibility constraints in aerial images. Incorporating higher resolution or ground-level datasets could extend this functionality to smaller species in the future. The optimal pathfinding module proved valuable in planning routes with minimal ecological disruption. By combining vegetation masking and spatial distance, the system generated environmentally aware paths that could support forest road planning or ranger routing. Overall, the modularity and flexibility of the system allow for further extension and regional customization. Future enhancements may include satellite image integration, fine-grained species identification using convolutional neural networks (CNNs), and expanded datasets covering diverse forest types in India. The project shows strong potential as a scalable, semi-automated tool for forest authorities, researchers, and conservation agencies. [5]

Conclusion

The Intelligent Tree Enumeration and Forest Analysis System for Environmental Monitoring leverages machine learning and structured datasets to revolutionize forest monitoring. By automating tree counting, species classification, green cover estimation, and optimal pathfinding, it enhances accuracy and efficiency over traditional methods. The integration of historical data analysis provides critical insights into forest trends, supporting sustainable management and conservation efforts. This system empowers researchers, policymakers, and environmental agencies with data-driven solutions to combat deforestation and promote ecological balance. By bridging technology and sustainability, it ensures that forests are preserved, monitored, and managed efficiently for future generations.

Acknowledgements

We express our sincere appreciation to Bharati Vidyapeeth College of Engineering for Women, Pune, for providing us with the necessary infrastructure, technical support, and encouragement throughout for successful execution of this project. Furthermore, we would like to thank our faculty members, and all those who contributed their time and expertise to help us achieve this milestone. Their valuable insights and continuous support played a crucial role in shaping our research. Lastly, we acknowledge the efforts of our entire project team, whose dedication and collaboration made this research a success.

References

[1]. Lujin Lv et al., UAV-Based Intelligent Detection of Individual Trees in Moso Bamboo Forests with Complex Canopy Structure, 2023.

[2]. Sumona Akter Shimu et al., NDVI Based Change Detection in Sundarban Mangrove Forest Using Remote Sensing Data, 2019.

[3]. Takeshi Motohka, Kenlo Nishida Nasahara, Hiroyuki Oguma, Satoshi Tsuchida, "Applicability of Green-Red Vegetation Index for Remote Sensing of Vegetation Phenology," Remote Sensing, 2010

[4]. André Coy, Dale Rankine, Michael Taylor, David C. Nielsen, Jane Cohen, "Increasing the Accuracy and Automation of Fractional Vegetation Cover Estimation from Digital Photographs," Remote Sensing, 2016.

[5]. Haoran Lin, Xiaoyang Liu, Zemin Han, Hongxia Cui, Yuanyong Dian, "Identification of Tree Species in Forest Communities at Different Altitudes Based on Multi-Source Aerial Remote Sensing Data," Applied Sciences, 2023.

[6]. Lin, H., Liu, X., Han, Z., Cui, H., Dian, Y. "Identification of Tree Species in Forest Communities at Different Altitudes Based on Multi-Source Aerial Remote Sensing Data." Applied Sciences, 13(4), 2023.

[7]. Katoh, M., Gougeon, F.A. "Improving the Precision of Tree Counting by Combining Tree Detection with Crown Delineation and Classification on Homogeneity Guided Smoothed High Resolution (50 cm) Multispectral Airborne Digital Data." Remote Sensing, 4(7), 1411-1438, 2012.

[8]. Parsakhoo, A., & Jajouzadeh, M. (2016). Determining an optimal path for forest road construction using Dijkstra's algorithm. Journal of Forest Science, 62(6), 264-268. DOI: 10.17221/9/2016-JFS.