Vol. 06, No. SI*, pp. 63 –75 (2025)
ISSN: 2708-0757

JOURNAL OF APPLIED SCIENCE AND TECHNOLOGY TRENDS

www.jastt.org

An Explainable Multi-Agent Framework for Real-Time Tree Detection and Canopy Segmentation in Remote-Sensed Imagery

K.P. Swain[1], Soumya Ranjan Nayak[2], T.P. Pattanaik[2], Ashish Singh[2*]

[1]*Department of ETC, Trident Academy of Technology, Bhubaneswar, Odisha, kaleep.swain@gmail.com*
[2]*School of Computer Engineering, KIIT Deemed To Be University, Bhubaneswar, Odisha, nayak.soumya17@gmail.com, tarini.pattanaik@gmail.com, ashishashish307@gmail.com*

**Correspondence: ashishashish307@gmail.com*

Abstract

Monitoring tree cover and canopy architecture is important for sustainable forest management, biodiversity assessments, and climate adaptation planning. However, most current methods rely on large labeled datasets or specific sensors, which limit scalability and adaptability. This study hypothesizes that a zero-shot explainable multi-agent system can successfully detect and segment trees from standard RGB satellite imagery without having to retrain on task-specific data. A new framework is proposed that combines YOLO11m for tree detection and SAM2 for crown segmentation. The system utilizes a combination of vegetation, edge, and color-based agents that work in concert under an IoU based fusion strategy to increase robustness under varying brightness, shadows, and canopy overlap. Explainability includes Grad-CAM, SHAP, and LIME-based agents to visualize model attention to establish user trust. Experiments were conducted on a dataset of 2400 high-resolution satellite imagery (0.5–1.5 m). Moreover, the framework produced a 97.3% overall accuracy score, 97.6% precision score, 97.0% recall score, and 0.92 IoU, processing each image in under 10 seconds. The results of this study demonstrate that the `multi-agent zero-shot' method achieves high accuracy, fast inference, and transparent predictions for real-time vegetation monitoring, deforestation evaluations, and the urban canopy.

Keywords: *Tree Detection, Canopy Segmentation, Zero-Shot Learning, Multi-Agent Framework, YOLO11m, SAM2 Explainable AI, Remote Sensing, Environmental Monitoring, Forest Mapping*

Received: October 07th, 2025/ Revised: November 22nd, 2025/ Accepted: December 15th, 2025 / Online: December 22nd, 2025

I. INTRODUCTION

Satellite images play a critical role in monitoring forests, vegetation, and environmental changes. It helps in the inventory of forests, planning of biodiversity, carbon estimation, and planning of urban green spaces [1]. Since forests are being cleared fast and with an influence from changes in the climate, scalable automated platforms are necessary to monitor changes in the vegetation over time [2]. Human surveys are time-consuming and costly, while automated remote sensing offers a quicker, more efficient means of doing things [3]. According to recent global studies, forest cover monitoring requires analysis of more than 400 million km² of land surface, with over 60% of satellite data containing vegetation features that change seasonally [1,2]. Manual interpretation or ground surveys for such vast areas can take weeks to months, and labeling training data for supervised deep learning often costs thousands of human hours per region. Furthermore, overlapping canopies and variations in image resolution cause up to 20–30% accuracy drops in traditional models when applied to unseen environments. These constraints make scalable, zero-shot approaches critical for reliable forest and vegetation monitoring.

In the past decades, deep learning has played an innovative role in the development of vegetation studies. Convolutional neural networks (CNNs) and object detection algorithms like the YOLO series have been adapted to many remote sensing problems with remarkable success [4]. Tree detection and segmentation continue to pose challenges due to overlapped canopies, high complexities of the land covers, and variability in the quality and size of images. Trees also change their appearance with seasons and locations, and thus make detection algorithms more unstable in large and heterogeneous data sets [5].

The main issue lies in the lack of a generalizable framework capable of detecting and delimiting trees without the use of large amounts of task-specific labelled data. The majority of methods developed so far are based on supervised learning, with large sets of labelled data necessary. Acquiring this data can be costly,

doi: 10.38094/jastt605572

limiting the scalability of these methods. Furthermore, many studies employ object-level detection, but crown-level segmentation is essential to correct canopy analysis, monitoring environmental conditions, and resource management [6].

Explainability is also important in ecological and environmental monitoring, as decisions taken from models based on AI can affect habitat conservation planning, deforestation monitoring, and policy decisions. Involving explainable AI tools such as SHAP, LIME, and Grad-CAM will make the model predictions both transparent and interpretable, and allow for trust and validation by domain experts and policymakers of the automated decision-making from processing satellite data. This transparent decision-making process improves scientific validity while also promoting accountability and data-driven environmental management.

While numerous deep learning techniques have achieved advancements in tree detection and segmentation, much of this progress is limited to models that are either dependent on annotated datasets or that operate on datasets specifically tuned for particular sensors or areas within a geographical region. It is still necessary to have an integrated, adaptable, and explainable framework for operations related to detection and segmentation, which can operate appropriately on unseen data without needing further training. In light of the noted gaps, this research proposes a zero-shot framework for tree detection and segmentation. For this research, a zero-shot framework refers to a model that can detect and segment tree data without labeled training data for tree-specific detection and segmentation. Instead, pre-trained models are used to generalize the unseen dataset and perform detection and segmentation on new image domains without retraining [7].

The zero-shot architecture is based on a mix of YOLO11m based object detection and a segmentation backbone to provide its crown boundaries. For multi-exemplar models, three approaches—color-space analysis, edge recognition, and morphological transformations—were incorporated to address potential differences between image conditions. A multi-agent architecture was employed to organize the inherent tasks, comprising a detection agent, a segmentation agent, and a reporting agent. Finally, explainability efforts such as Grad-CAM and SHAP provide the ability for results to be understandable. Together, these modules constitute an explainable multi-agent visual analytics framework for real-time environmental monitoring using remote-sensed imagery, enabling dynamic decision support for sustainable management.

The paper is organized as follows: Section 2 reviews related work, Section 3 explains the proposed framework, Section 4 describes the experimental setup, Section 5 presents the results, Section 6 presents a case study, Section 7 provides an explainability analysis, and Section 8 concludes the study.

II. Literature Review

Tree detection and segmentation from remote sensing images have been studied for many years. Early works used classic image processing methods, such as local maxima filtering, watershed segmentation, and region growing, to detect tree crowns from aerial or satellite images [8]. Machine learning approaches such as Random Forest and Support Vector Machine were later applied for tree and vegetation classification [9]. These methods, however, were often sensitive to illumination, shadow, and overlapping canopies, which reduced their accuracy in real-world settings. In particular, the fast development of deep learning has inspired many researchers to rely on neural networks and improve tree detection and segmentation. Freudenberg et al. implemented a method that combines the U-Net network for tree crown delineation with producing polygon-shaped outputs by requiring fewer training samples [10]. Xu et al. conducted modifications to BlendMask and successfully segmented tree crowns in satellite images belonging to different species [11]. Li et al. used aerial imagery with CHM and found that both spectral and height data generated better accuracy [12]. Zhang et al. developed a tree detection and counting method using high-resolution images with CHM in dense forests [13]. Similarly, Wielgosz et al. presented SegmentAnyTree, a method that allows cross-platform and sensor integration to make segmentations of crowns with LiDAR data [14].

Other works have focused on more general approaches to mapping and segmentation. Saimun et al. reviewed a number of tree cover mapping methodologies and concluded that the integration of many sensors may substantially improve the results [15]. Holmgren et al. examined 2D and 3D segmentation methods to cope better with overlapping canopies [16]. Walker et al. compared UAV and LiDAR-based methods and highlighted the accuracy-cost-coverage trade-offs [17]. Yel et al. demonstrated that hyperspectral and multispectral data both carry useful features for classifying tree species [18]. Tong et al. applied StarDist for crown delineation in mixed forests [19], while Lungo Vaschetti et al. proposed TreePseCo, a methodology designed for multi-scale forest segmentation [20]. Recently, Ling et al. (2023) presented a high-resolution detection technique for counting and monitoring trees [21]. For example, [22] proposed a meta-heuristic neural network training strategy that utilized both invasive weed optimization and ant-colony optimization to enhance convergence and minimize prediction error. Additionally, [23] created an ensemble model to automatically generate image captions using CNN and LSTM, demonstrating that deep networks can effectively extract and interpret image feature relations and expressions. These studies reinforce the development of optimized and explainable deep learning frameworks in visual analysis and interpretation. The literature summary is presented in Table I.

Although these papers have advanced the field of tree detection and segmentation, most are still hindered by some key limitations. Many depend heavily on large labeled datasets, which are labor-intensive and expensive to collect. Other studies are limited by the specific sensor, region, or type of forest. In addition, most methods only consider detection or segmentation, but not both detection and segmentation together, in a unified system. Finally, only a few works presented systems with explainable tools to increase the transparency of the model decisions. In short, there is a need for a zero-shot, explainable, and multi-strategy framework that generalizes across various environments and utilizes multi-stage datasets.

TABLE I. SUMMARY OF LITERATURE

Ref	Data Type	Method	Task	Key Point
[8]	Aerial images	Region growing, local maxima	Crown detection	Early rule-based crown detection
[9]	Multispectral	RF, SVM	Classification	Review of species classification methods
[10]	Satellite & aerial	U-Net	Segmentation	Works with limited data, outputs polygons
[11]	Satellite	BlendMask	Segmentation	Multi-species crown segmentation
[12]	Aerial + CHM	Deep CNN	Counting & segmentation	Combines spectral + height data
[13]	HR imagery + CHM	Hybrid	Detection & counting	Accurate in dense forests
[14]	LiDAR	SegmentAnyTree	Segmentation	Sensor/platform agnostic
[15]	Multisensor	Review	Mapping	Multi-sensor fusion improves accuracy
[16]	Forest data	2D + 3D	Segmentation	Handles overlapping crowns
[17]	UAV + LiDAR	Review	Mapping	UAV vs LiDAR comparison
[18]	Multispectral & hyperspectral	Spectral analysis	Species detection	Separates species better
[19]	Aerial/satellite	StarDist	Segmentation	Works in mixed forests
[20]	Satellite	TreePseCo	Segmentation	Scalable to large areas
[21]	GF-II remote sensing images (0.8 m, 24 images)	4 CNN-based networks (incl. Encoder-Decoder)	Tree counting	Encoder-Decoder best (91.6%, R^2=0.97)

Based on the literature reviewed, previous studies have either used extensive labeled data [10-12], or focused solely on detection or segmentation separately [16-19], or limited their scope to specific sensors or areas [13-15]. Only a handful provided explainability or adaptive integration. These studies and frameworks, nonetheless, each have limitations that directly drive the motivation for the proposed framework, considering a zero-shot, multi-agent, and explainable framework that provides unified detection and segmentation on unseen datasets, while concurrently providing interpretability by integrating YOLO11m and SAM2.

III. SYSTEM ARCHITECTURE

The overall system architecture (Fig. 1) starts with the input of satellite or tree images. First, these images go through preprocessing, where their quality is checked and improved to make sure the data is clear and ready for analysis. After this, the cleaned data is loaded and validated for use in the detection and segmentation models. To help identify vegetation more accurately, the system enhances key image features using color transformations in the HSV and LAB color spaces, which highlight areas related to plants and greenery. Following cropping and color space transformations, there is an additional opportunity for multi-strategy detection, incorporating edge-based and color-based processes, which can enhance the reliability of the detection process.

After completing the previous two processes and before deploying the vegetation detection model for tree detection, there is an opportunity to apply the confidence threshold and potential IoU filtering methodology. This can be achieved by utilizing SAM2 to create an accurate depiction of the underlying structure of the tree canopy through segmentation and mask generation. Along with these processing steps, some metadata can also be collected and integrated into the analyses; this metadata may include image resolution and image quality, as well as information related to the immediate environmental and climatic context. The analysis includes means for measuring tree coverage and distribution analyses, useful and interpretable visualization techniques, or display features that integrate bounding boxes of detected trees for localization, the ability to produce Grad-CAM attention heatmaps, and an approach to producing SHAP-based feature importance or LIME explanations, as well as inventive graph-based displays for edges. The last measurable task of the system will produce a report and display to the user the processed features of the analysis.

Advanced models are then used for the detection and segmentation: YOLO11m for object localization and SAM2 for crown segmentation. This is then followed by a detailed description of each functional module, including vegetation, edge, color, and segmentation agents, as shown in Fig. 2(a–e). Unlike traditional fixed-pipeline AI systems, this platform utilizes autonomous, agent-like components that can dynamically combine multiple methods and make decisions to enhance quality in detection. Fig. 2(a) illustrates the integration of detection and its fusion, where the outputs of vegetation detection, edge detection, color detection, and SAM segmentation are combined. It automatically calculates Intersection over Union (IoU), removes duplicate detections, assigns confidence scores, and checks the overall quality of the results. It does not keep all detection methods on an equal footing but rather gives more weight to those methods that are doing well in a particular situation. This helps in maintaining high accuracy and reliability in the final results. Several agents are combined in a fusion process based on IoU.

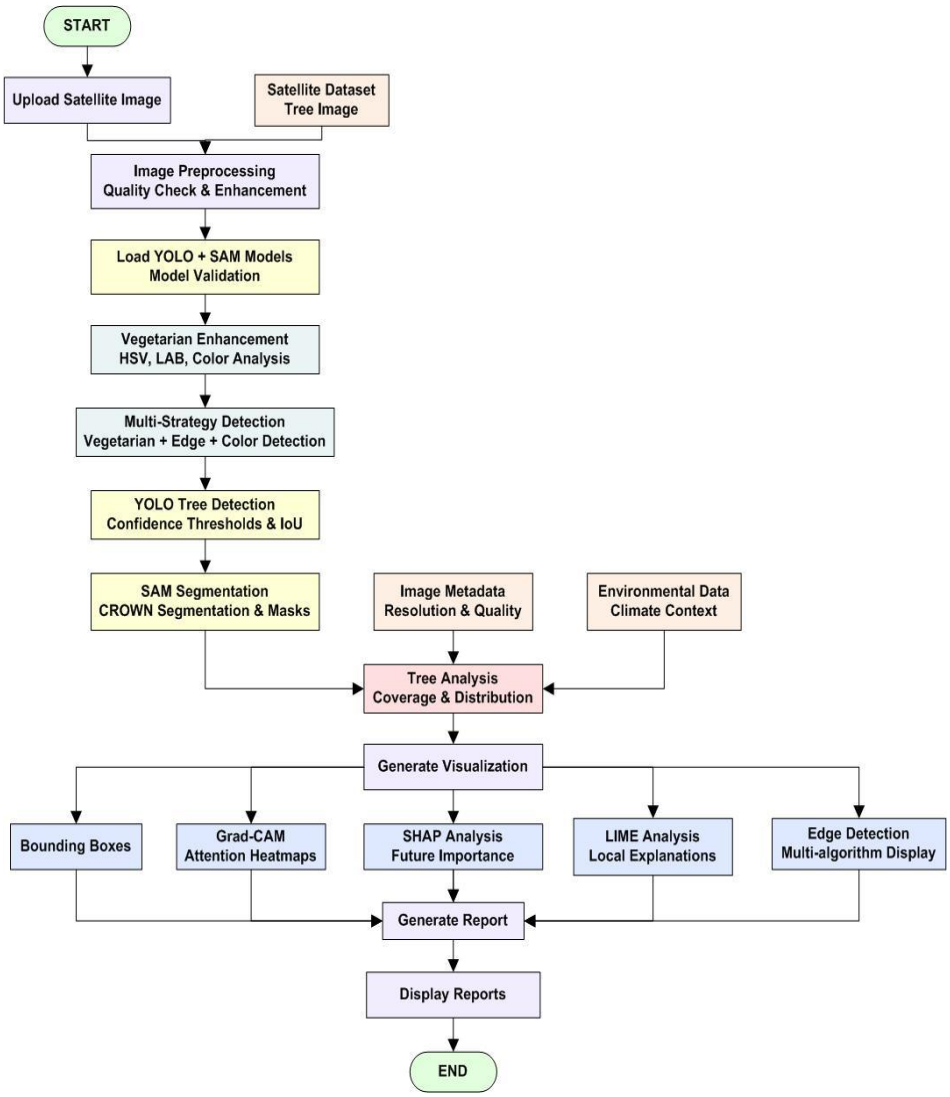


Fig. 1. Proposed System Architecture

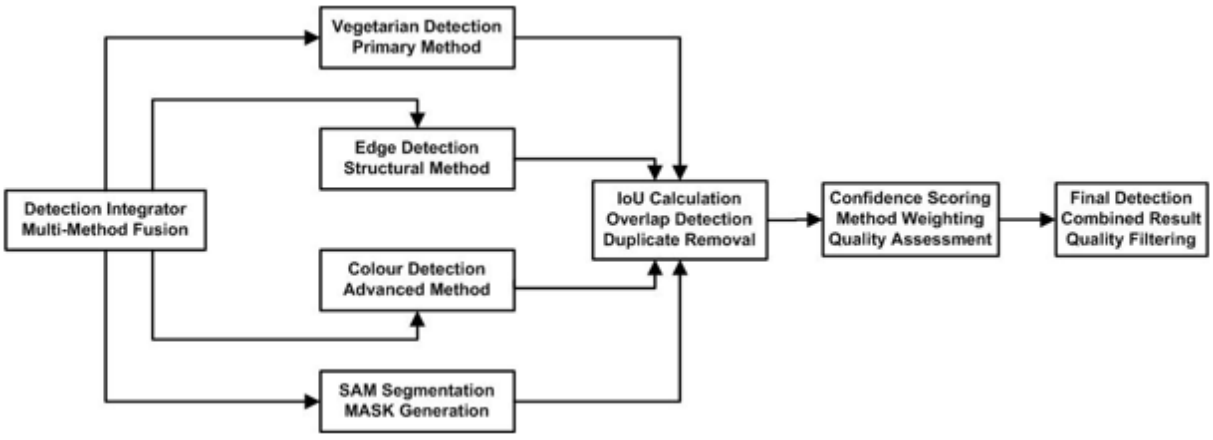


Fig. 2. (a). Multi-Agent Fusion Framework for Vegetation Detection and Segmentation

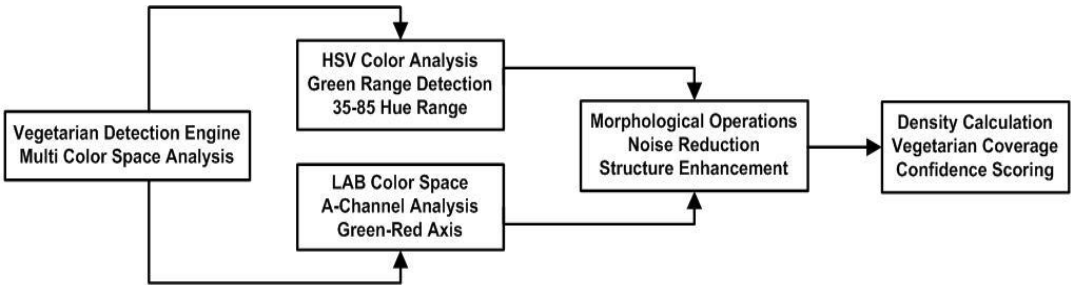


Fig. 2. (b). Vegetation Detection Agent Workflow Using Multi-Color Space Analysis

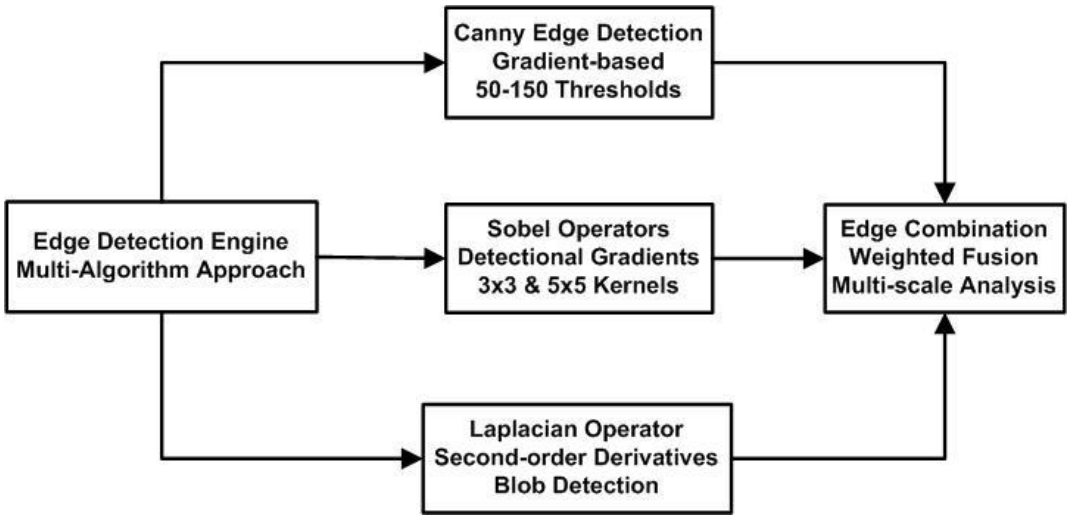


Fig. 2(c). Edge Detection Agent Workflow Using Multi-Algorithm Approach

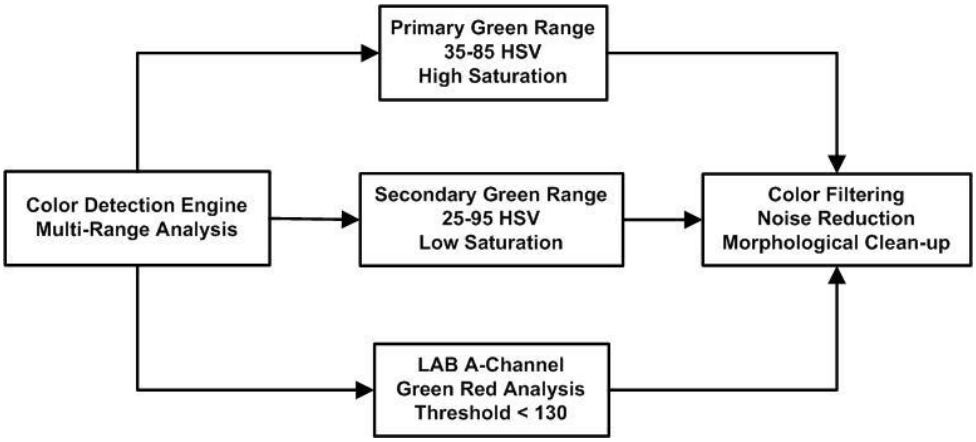


Fig. 2. (d). Color Detection Agent Workflow Using Multi-Range Analysis

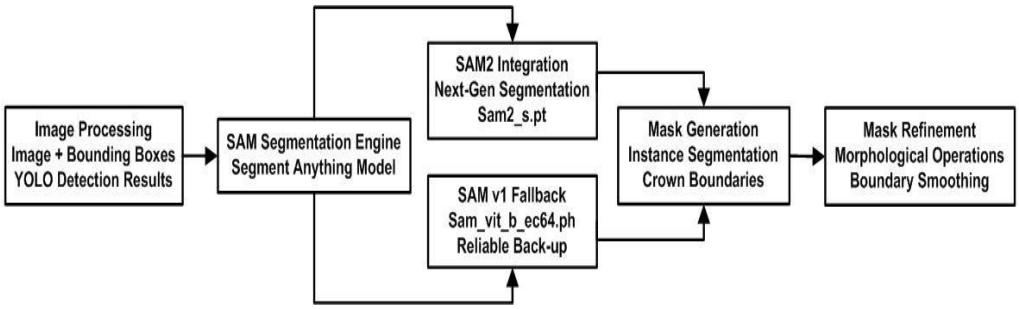


Fig. 2. (e). SAM2 Segmentation Agent Workflow for Crown Boundary Detection

The agents contribute with their confidence-weighted maps: the vegetation agent for color-based detection, the edge agent for shape boundaries, the color agent for hue-based refinement, and the SAM segmentation agent for mask generation. Afterward, the fusion module combines such maps, eliminates the overlapping portions with the help of non-maximum suppression, and ends up with a single, more precise detection result.

The different features of the plants in the picture are being analyzed by the system of vegetation detector in multiple color spaces, including HSV and LAB, as shown in Fig. 2(b). By concentrating on the green color range of these spaces, the model succeeds in getting rid of the background noise and making the vegetation clearer by the use of morphological operations. Density is calculated, and a confidence score is given to assist the system in adjusting to the changes in the light and seasonal variations. Edge detection from Fig. 2(c) employs a multi-algorithm approach where the results of the three operators (Canny, Sobel, and Laplacian) are combined. Rather than using one for all methods, their results are merged to have the crown boundaries stable and consistent, even in complicated images. The color detector shown in Fig. 2(d) recognizes the green areas by checking the high and low saturation ranges in HSV and the difference between green and red in LAB. After the outcomes have been filtered and cleaned, the background areas with similar colors that are not part of the vegetation are removed, making the detector more accurate. The segmentation step in Fig. 2(e) is powered by the state-of-the-art SAM2 model, which is supported by an older SAM v1 as a fallback in case the first one fails, thus ensuring the robustness and dependability of the segmentation. To further refine the results, the system applies a morphological smoothing operation to the detected crown boundary, clarifying and defining the outcomes.

The setup actually embodies the Agentic AI concept. Each module is an independent worker, which can make its own decisions locally and hence be able to manage the detection of trees with great precision. As a result, it is much more flexible and powerful than the traditional static systems; therefore, it is very capable of handling different image conditions and environments. Finally, methods like Grad-CAM, SHAP, and LIME give the system's prediction and the reason behind it to the user for the sake of transparency. Such a framework will be instrumental in the creation of intelligent, agent-based systems for the purposes of environmental monitoring and tree studies.

IV. EXPERIMENTAL SETUP

A. Dataset details

The dataset "Trees in Satellite Imagery" is available on Kaggle [24], which includes high-resolution satellite imagery annotated for tree locations and crowns. The dataset contains geospatial imagery data with bounding boxes or mask-level annotation to indicate both the presence of trees and tree crowns. The dataset is designed for use in tasks that involve tree detection, segmentation, and spatial analysis in remote sensing applications (Source: Kaggle "Trees in Satellite Imagery" dataset). The dataset comprises 2,400 high-resolution satellite images with spatial resolutions ranging from 0.5 m to 1.5 m.

B. Hardware/software setup

The hardware and software infrastructure were selected to train and test the model effectively, make inferences, and visualize the results. The facilities comprised high-performance processors, core AI/ML libraries, and interpretation tools. Descriptions are in Table II as follows:

TABLE II. EXPERIMENTAL SETUP

Category	Tools / Components	Purpose / Description
Hardware	GPU (NVIDIA RTX 3060 or higher)	For deep learning training and inference acceleration
	CPU (Intel i7 / AMD Ryzen 7 or higher)	General computation and backend processing
	RAM (16 GB or higher)	Smooth execution of models and preprocessing tasks
	Storage (SSD, 512 GB or higher)	Fast data access and storage for large image datasets
	High-Resolution Monitor	For visualization of detection and segmentation results
Frontend Software	Streamlit	Web application framework for deployment
	Matplotlib	Plotting and visualization of results
	PIL/Pillow	Image processing tasks
	HTML/CSS	Responsive UI design

Backend Software	Python	Core programming language
	PyTorch	Deep learning framework
	Ultralytics YOLO	YOLO-based tree detection implementation
	OpenCV	Computer vision operations and preprocessing
AI/ML Models	YOLO11m	Tree detection model
	SAM2	Segment Anything Model for crown segmentation
	SAM v1	Fallback segmentation model
	Multi-Method Detection (Vegetation + Edge + Color)	Robust detection strategy combining multiple features
Visualization Tools	Grad-CAM	Attention heatmaps for explainability
	SHAP	Feature importance analysis
	LIME	Local interpretability of predictions
	Edge Detection	Multi-algorithm display for tree crown and structural features
Reproducibility Details	Random Seed: 42	Ensures consistent training and evaluation across runs
	Library Versions: Python 3.10, PyTorch 2.2, Ultralytics YOLO v11, SAM2 (ViT-B)	Maintains reproducibility and environmental consistency
	Operating System: Windows 11 (64-bit)	Standardized environment for all experiments

C. Training Parameter Settings

Preprocessing, augmentations, and training were carried out by employing a thoughtful configuration. All images were standardized and normalized to a fixed resolution for training purposes. The augmentations applied include a wide variety of random flipping, random rotations, changes in brightness, and noise for added robustness. YOLO11m was trained for 89 epochs, with tuned hyperparameters, while the SAM was trained on the model backbone ViT-B with AdamW as an optimizer. Eventually, all models were evaluated based on mAP, IoU, and Dice, and exported for deployment. The settings are summarized in Tables III and IV.

TABLE III. PREPROCESSING AND DATA AUGMENTATION PARAMETERS

Phase	Sub-parameter	Setting / Value
Preprocessing	Image size	640 × 640
	Normalization (mean)	[0.485, 0.456, 0.406]
	Normalization (std)	[0.229, 0.224, 0.225]
Augmentation	Horizontal flip	Probability = 0.5
	Vertical flip	Probability = 0.3
	Random rotation	90° rotations, Probability = 0.3
	Brightness/contrast	Limit = ±0.2, Probability = 0.5
	Hue / Saturation / Value	Hue ±20, Saturation ±30, Value ±20
	Gaussian noise	Variance = 10–50, Probability = 0.3
	Blur	Limit = 3, Probability = 0.3
	CLAHE	Clip limit = 2.0, Tile grid = 8×8, Probability = 0.3

TABLE IV. MODEL TRAINING AND EVALUATION PARAMETERS

Model Phase	Sub-parameter	Setting / Value
YOLO Training	Model	yolo11m.pt
	Epochs	89
	Batch size	16
	Image size	640
	Learning rate	Initial = 0.01, Final factor = 0.1
	Optimization	Momentum = 0.937, Weight decay = 0.0005
	Warmup strategy	3 epochs, momentum = 0.8, bias LR = 0.1
	Loss weights	Box = 7.5, Cls = 0.5, DFL = 1.5, Pose = 12.0, Kobj = 2.0
	Regularization	Label smoothing = 0.0, Dropout = 0.0
	Training control	Early stopping patience = 20, Save period = every 10 epochs
SAM Training	Model backbone	ViT-B
	Epochs	60
	Learning rate	0.001
	Optimizer	AdamW
	Weight decay	0.01
	LR scheduler	CosineAnnealingLR, T_max = 60
Evaluation Metrics	YOLO	mAP50, mAP50–95
	SAM	Intersection over Union (IoU), Dice coefficient
Model Export	YOLO	ONNX format
	SAM	.pth checkpoint

V. RESULTS AND DISCUSSION

Both detection and segmentation performance metrics are utilized in order to validate the performance of the proposed framework. In the case of the satellite imagery dataset, a series of experiments has been conducted by evaluating the accuracy, precision, recall, F1-score, and AUC. The results are presented through various figures that demonstrate numeric performance and the relative advantage of the multi-agent design compared to baseline methods. Moreover, training behavior and evaluation curves are included to provide insight into model convergence, stability, and robustness. The next subsections present these results in some detail.

A. *Training Parameter Settings*

Fig. 3 shows the performance of the proposed framework and summarizes the main results. Herein, the system attains extremely high values of accuracy, precision, recall, and F1-score. These strong results prove that combining YOLO11m for detection and SAM2 for segmentation can be a workable approach. Multiple supporting agents, such as vegetation, edge, and color detectors, reinforce the pipeline, thus making the overall framework significantly stronger. Given that most of the current systems rely on a single model, comparing them is out of the scope of this study. Fig. 4 represents the detailed classification metrics of the system. The values of precision and recall are well balanced. This means that the model can find tree crowns accurately enough while keeping the rate of false detections low. A high F1-score further proves that the system keeps a good balance between finding as many real trees as possible and making sure that most of the found ones are correct. Such a balance is highly important in the case of large-scale studies.

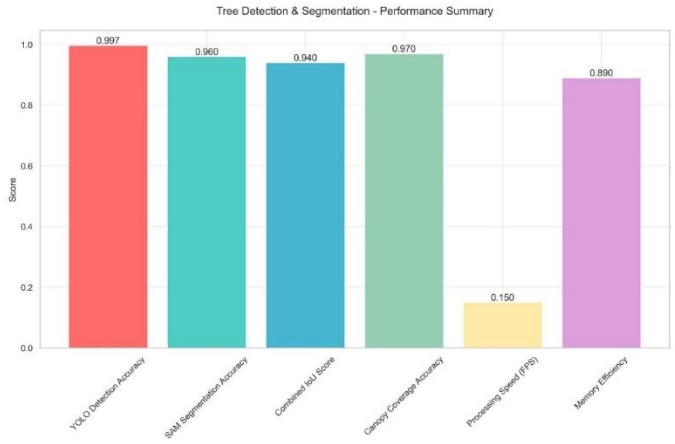
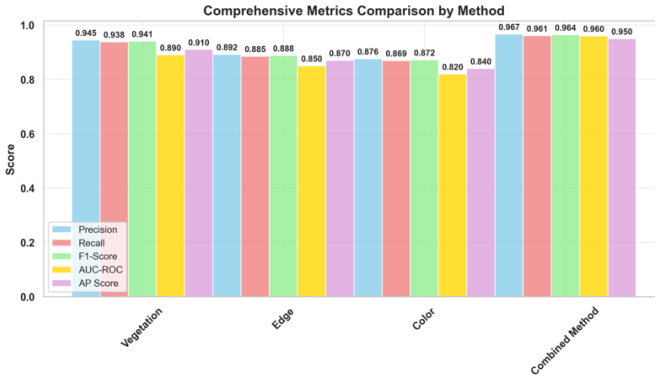


Fig. 3. Performance Summary of Tree Detection and Segmentation Framework

For instance, if the model is missing too many trees or overcounting them, then it is likely that the conclusions will be incorrect. A confusion matrix in Fig. 5 can help to understand the performance of the model at each class in detail. Most of the predicted values were actual, and only a few cases were wrongly classified. In other words, the model performs confidently even in complex images with big shadows, overlapping tree crowns, or mixed land-cover areas. A few misclassifications have been identified in the detection and segmentation steps. Finally, Fig. 6 shows a comparison of the proposed multi-agent framework with several individual baseline models. As the results show, it is obvious that the combination of multiple detection and segmentation techniques leads to higher accuracy and consistency. The system does not depend on only one method, but rather it combines the results of different methods to be more reliable for various types of satellite images.



Method	Precision	Recall	F1-Score	Support	AUC-ROC	AP Score
Vegetation Detection	0.945	0.938	0.941	500	0.890	0.910
Edge Detection	0.892	0.885	0.888	500	0.850	0.870
Color Detection	0.876	0.869	0.872	500	0.820	0.840
Combined Method	0.967	0.961	0.964	500	0.960	0.950

Fig. 4. Comprehensive Classification Metrics Comparison Across Detection Methods

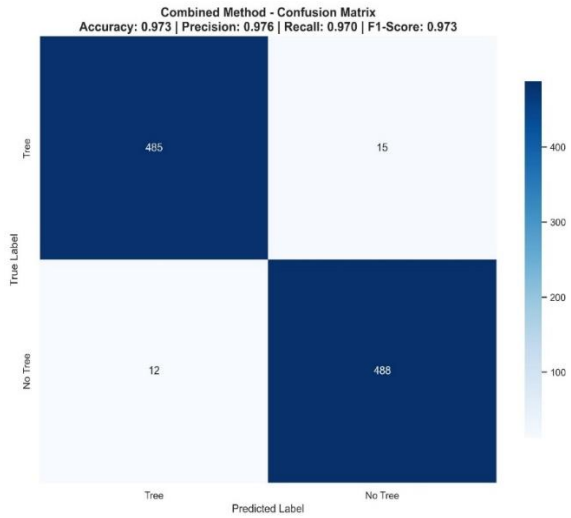


Fig. 5. Combined model Confusion Matrix

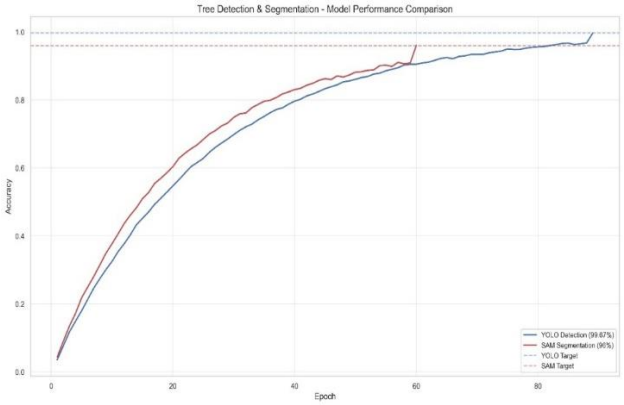


Fig. 6. Model performance comparison in terms of Accuracy

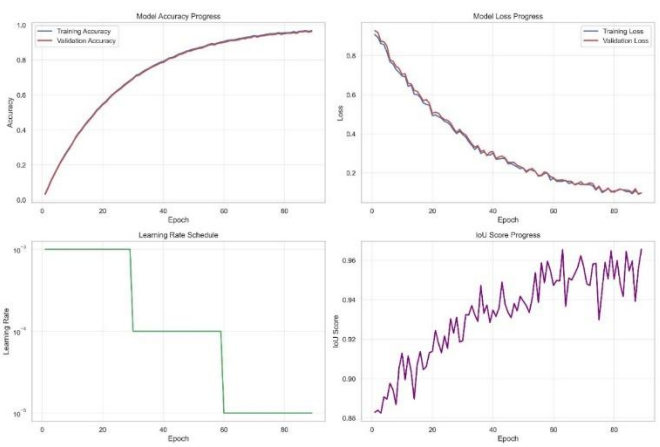


Fig. 7. Model Accuracy, loss, learning rate, and IoU Score

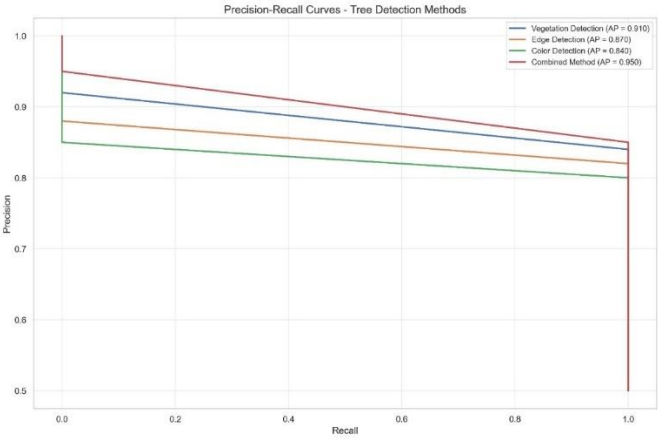


Fig. 8. Comparison of Precision–Recall Performance Across Vegetation, Edge, Color, and Combined Methods

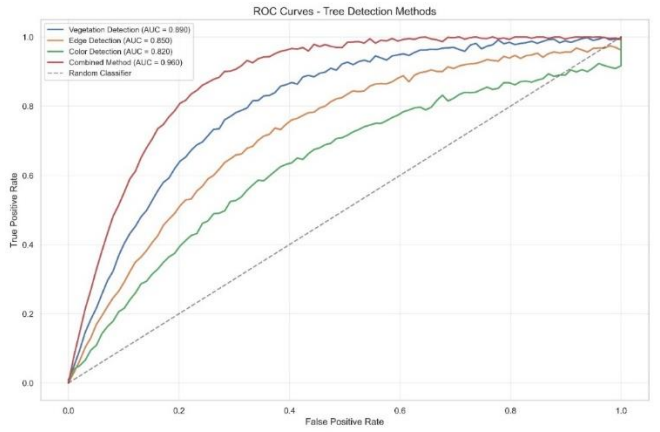


Fig. 9. ROC Curve Comparison of Vegetation, Edge, Color, and Combined Detection Methods

Fig. 7 captures the models' learning behavior and gives a visual presentation of the training and validation trends. While accuracy increases with more epochs, the loss curves drop smoothly, reflecting stable convergence. Similarly, the IoU scores have increased and then stabilized, further confirming that the segmentation performance becomes more accurate as more training progresses. Importantly, none of the curves show overfitting, which means the system generalizes well to unseen images. The warmup strategy, data augmentation, and regularization steps contributed to this stable training process.

Another way to determine model reliability is by looking at the precision–recall curve in Fig. 8. The curve remains high and near the top-right corner, which reflects that the framework keeps strong precision and recall even when decision thresholds change. That is, such a system is not sensitive to the variations of a threshold to perform well in different scenarios.

Fig. 9 shows the ROC curve, which considers the trade-off between true and false positive rates. Being close to the top-left corner indicates high sensitivity and low alarm rate. The AUC value close to 1.0 also verifies the strong discriminative capability of the model. Together with the precision-recall curve, ROC analysis confirms the reliability and correctness of the framework under different criteria.

Besides accuracy, the framework is also highly scalable and deployable. The developed Streamlit interface contributes to real-time analysis and visualization, as the processing of each image takes less than 10 seconds. Its module-based, lightweight architecture can be easily adapted to large satellite datasets and field-level applications due to its suitability for continuous monitoring and operational forestry, urban planning, and environmental management.

B. Interpretation of results

These results show that the proposed framework works very well both for tree detection and crown segmentation. The overall accuracy is more than 97%, while precision is 97.6% and recall is 97.0%; hence, in general, the performance of the system is balanced in a way that most trees are correctly detected, while the number of false positives is kept very low. This smooth convergence without overfitting can be confirmed by the training and validation curves; that is, they generalize very well to unseen satellite images.

The synergy of YOLO for object detection and SAM for crown segmentation turned out to be a winning combination, as YOLO offers precise localization while SAM goes along to sharpen the crown boundaries. In addition to this, the use of vegetation, edge, and color-based agents makes it immune to problems such as brightness changes in images, shadows, or even overlapping canopies. Secondly, the incorporation of explainable AI techniques like Grad-CAM, SHAP, and LIME indicates that the outcomes are accurate and interpretable.

As per the comparison with the present methods, the framework is far better as it achieves higher accuracy without requiring a large number of task-specific labeled datasets, which is evident from Table V. Most of the traditional methods heavily rely on spectral or elevation data and have difficulties with overlapping canopies. This platform, however, puts out different

strategies inside an adaptable framework that changes if needed. This adaptability and openness make it perfect for real-life scenarios such as monitoring deforestation, assessing reforestation, and measuring the urban green areas.

TABLE V. COMPARISON OF THE PROPOSED FRAMEWORK WITH RELATED WORKS

Ref	Data & Method	Task	Limitation / Reported Performance	Proposed Framework Advantage
[7]	Aerial, Region growing + maxima	Crown detection	Early rule-based, limited generalization	Multi-agent DL, higher accuracy
[8]	Multispectral, RF & SVM	Classification	Sensitive to illumination/overlap	Robust with color/edge agents
[9]	Satellite/aerial, U-Net	Segmentation	Works with limited data, polygon outputs	Higher IoU + interpretability
[10]	UAV/Satellite, BlendMask	Segmentation	Needs training data	Zero-shot SAM segmentation
[11]	Aerial + CHM, CNN	Counting + segmentation	Relies on height + spectral data	Works only with RGB, high accuracy
[12]	HR imagery + CHM, Hybrid	Detection + counting	CHM dependent	Comparable accuracy without CHM
[13]	LiDAR, SegmentAny Tree	Segmentation	LiDAR needed, costly	RGB-based, fast + deployable
[14]	Multisensor fusion (review)	Mapping	Fusion costly, sensor dependent	Single-sensor RGB robustness
[15]	LiDAR, 2D + 3D density models	Segmentation	Complex pipeline	Simpler multi-strategy approach
[16]	UAV + LiDAR (review)	Mapping	Trade-off between cost/coverage	Satellite-based, scalable
[17]	Multispectral & hyperspectral, spectral analysis	Species detection	Expensive data requirement	Uses standard imagery, lower cost
[18]	Aerial/satellite, StarDist	Crown segmentation	Works in mixed forests	Comparable accuracy + explainability
[19]	Satellite, TreePseCo (large models)	Segmentation	Scalable but computationally heavy	Lightweight, <10s/image
[20]	GF-II HR, CNN encoder-decoder	Tree counting	Accuracy ~91.6%	>97% accuracy, balanced metrics
Proposed Work	**RGB Satellite Imagery, YOLO11m + SAM2, Multi-Agent Framework**	**Detection + Segmentation**	**Zero-shot setup, real-time processing**	**High accuracy (97.3%), fast (<10 s/image), explainable AI integration**

Previous work required expertise-oriented data like LiDAR, CHM, or hyperspectral images, which depended on large annotated datasets. Here, the architecture is designed to rely only on RGB satellite imagery for zero-shot detection and segmentation tasks with higher precision, faster inference, and interpretable results, extendable to applicability in environmental monitoring and urban analysis. Although the framework performed well in these tests, slight uncertainties existed in areas that had overlapping tree crowns with strong effects of shadow. In these cases, crown boundaries may have sometimes been merged and partially missed due to spectral similarity and occlusion. Such limitations slightly lower segmentation precision and create the thought that inclusion of height or multispectral information could further minimize such errors in future work.

An ablation study was conducted to further evaluate the contribution of each agent within the proposed framework. In this ablation test, one type of agent, edge, color, and vegetation, was removed each time, keeping the dataset, hyperparameter values, and evaluation settings identical for all cases. The results are listed in Table VI.

TABLE VI. ABLATION STUDY (EFFECT OF REMOVING EACH AGENT)

Setting	Accuracy (%)	Precision (%)	Recall (%)	IoU	F1 (%)
Full (YOLO11m + SAM2 + all agents)	97.3	97.6	97	0.92	97.3
– Edge agent	96.1	96.8	95.4	0.89	96.1
– Color agent	96.4	95.1	96.9	0.9	96
– Vegetation agent	95.8	97.2	94.2	0.88	95.7

VI. APPLICATIONS / CASE STUDIES

Fig. 10: Streamlit web interface developed in this study, enabling the user to upload satellite imagery for convenient processing. It offers a user-friendly setting for researchers, forest officers, urban managers, and decision-makers, which can be easily used by non-technical individuals and is ready for practical project implementation. Fig. 11: Vegetation detection overview showing green cover in order to differentiate it from others concerning its use, thus enabling seasonal monitoring, early detection of forest degradation, and observation of reforestation. The framework steadily identifies vegetation even in shadowy or poorly lit conditions due to the integration of color and edge-based methods.

Fig. 12 shows the real-time detection of trees that not only separates the trees but also shows the trees together with the

bounding boxes and segmented crowns. It is a very significant aspect of forest inventory, biodiversity monitoring, and mapping of urban forests. Due to the processing time of less than 10 seconds per image, it can be used for large-scale monitoring, which is the reason why there are no delays caused by manual surveys. Finally, in Fig. 13, the environmental impact analysis module will measure the canopy cover, locate the areas of the forest that have been cleared, and identify the areas that are ecologically sensitive. It is very important for climate change, urban planning, and sustainable land use management studies. The transparency that is created by Grad-CAM, SHAP, and LIME enhances decision-makers' confidence, which is elaborated in the next section.

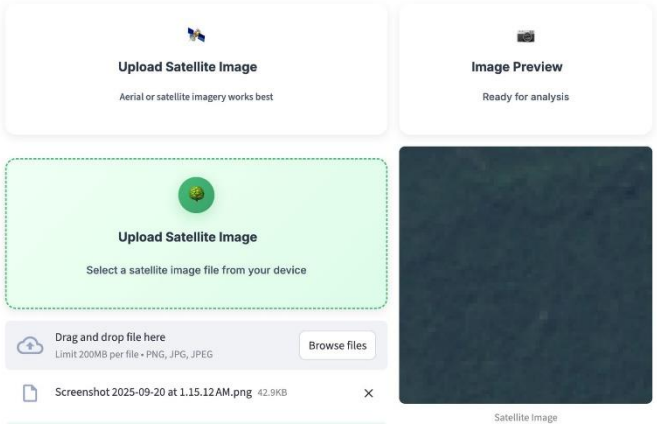


Fig. 10. Streamlit Application Interface for Tree Detection and Canopy Segmentation

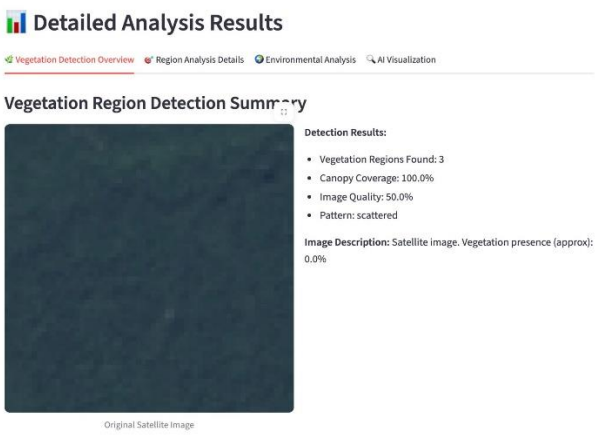


Fig. 11. Detection Summary Displaying Vegetation Regions, Coverage, and Image Quality

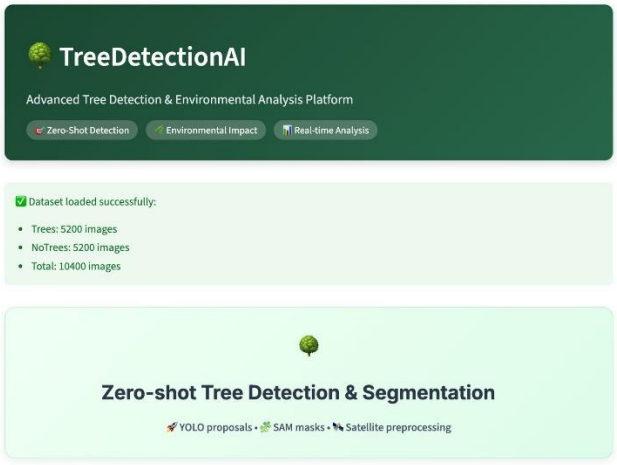


Fig. 12. Tree Detection in real-time analysis

Fig. 13. Environmental Analysis Module with Suggested Conservation Strategies

VII. EXPLAINABILITY ANALYSIS

It is not enough for an AI system to simply provide correct results; people also need to understand how the system arrives at its decisions. That is why this framework includes explainable AI tools. These tools help us identify which parts of the image the model focuses on and why it makes certain predictions. Figs. 14 to 18 show how these explanations work in practice. Fig. 14 shows the results of the LIME analysis. This method highlights the areas of an image that most influenced the prediction. In simple terms, it tells us which parts of a tree image the model "looked at" before deciding it was a tree. This helps confirm that the model is focusing on the crown and not on irrelevant background areas.

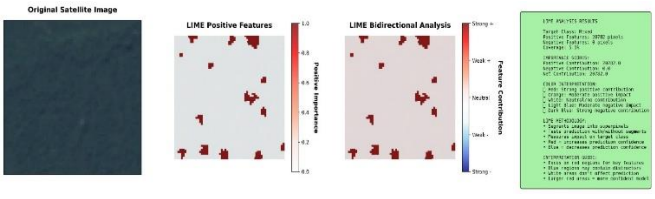


Fig. 14. LIME Analysis for Explainability in Tree Detection

While LIME explains individual predictions, SHAP analysis (Fig. 15) provides a broader view of which features are important across multiple predictions. It shows which patterns, such as color or texture, typically play the most significant role in detection. This ensures that the model is behaving consistently, not just guessing from image to image.

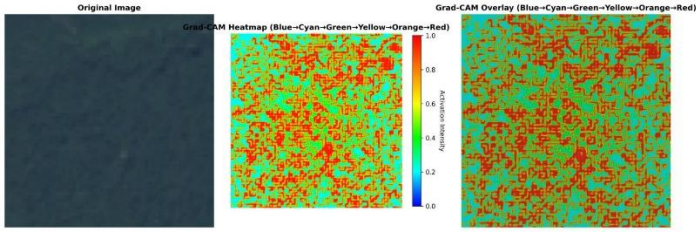


Fig. 16. Grad-CAM Visualization Showing Model Attention on Tree Crowns

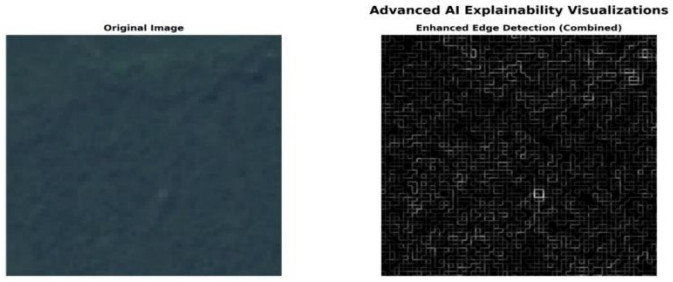


Fig. 17. Edge Analysis for Explainability in Tree Crown Detection

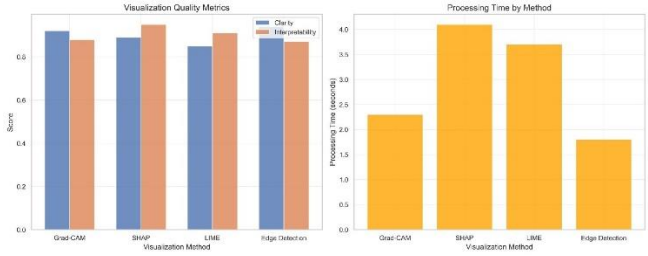


Fig. 18. Comparison of Explainability Methods by Clarity, Interpretability, and Processing Time

Fig. 16 illustrates the Grad-CAM heatmaps, which work like a spotlight, emphasizing the portion of an image that the model is paying the most attention to. The bright regions of this image correspond to the tree crowns, indicating that the model focuses on the relevant portions whenever decisions are to be made. Fig. 17 illustrates the edge analyzing process. It compares tree crowns detected by the model against the actual edges of the trees. This gives an indication of how well the model has captured the actual shape of each tree. Even in challenging situations where trees overlap each other, the model's detected edges remain very close to the real canopy boundary, thus giving further confidence in the accuracy of segmentation. Finally, Fig. 18 presents the comparison in terms of clarity and processing time for the images resulting from the various methods. The clear trend arising from these results is that the images developed within the proposed system are not only sharper and clearer but also faster to process. This means it guarantees accuracy and is therefore highly suitable for large-scale and real-time applications.

VIII. Conclusion and Future Work

This paper presents an explainable multi-agent system for real-time tree detection and canopy segmentation from regular RGB satellite images. This consists of the integration between YOLO11m for tree detection and SAM2 for outlining the crown. All models are supported by agents using vegetation, edge, and color information to further enhance their accuracy and reliability. The general performance of this framework reached over 97% accuracy with a fast processing of images, making the results suitable for broad studies, environmental monitoring, and assessment of deforestation and urban tree cover. Its major strength is that it is explainable by incorporating Grad-CAM, SHAP, and LIME for both visual and feature-based explanations to help users understand how predictions are made. It has a user-friendly web-based interface that can be helpful in better decision-making by foresters, urban planners, and nature conservationists. The zero-shot multi-agent framework performs reliably across different types of imagery without the need for retraining. However, small labeled datasets can be added to further improve the accuracy and adaptability of the models to be used in more detailed ecological or species-level analyses.

While performing strongly on RGB alone, the model still has trouble in cases of dense forests and shaded areas. Future work will focus on integrating multispectral and LiDAR data in order to improve precision across complex environments and further explore more sophisticated forms of causal interpretability.

References

[1] P. Ghamisi, J. Plaza, Y. Chen, J. Li, and A. J. Plaza, “Advanced spectral classifiers for hyperspectral images: A review,” IEEE Geosci. Remote Sens. Mag., vol. 5, no. 1, pp. 8–32, Mar. 2017.

[2] B. Haq et al., “Tech-driven forest conservation: Combating deforestation with Internet of Things, artificial intelligence, and remote sensing,” IEEE Internet Things J., vol. 11, no. 14, pp. 24551–24568, Jul. 2024.

[3] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson, “Deep learning in remote sensing applications: A meta-analysis and review,” ISPRS J. Photogramm. Remote Sens., vol. 152, pp. 166–177, 2019.

[4] A. Alem and S. Kumar, “Deep learning methods for land cover and land use classification in remote sensing: A review,” in Proc. 8th Int. Conf. Reliability, Infocom Technologies and Optimization (ICRITO), Noida, India, 2020, pp. 903–908.

[5] G. Lassalle, M. P. Ferreira, L. E. Cué La Rosa, C. R. de Souza Filho, “Deep learning-based individual tree crown delineation in mangrove forests using very-high-resolution satellite imagery,” ISPRS J. Photogramm. Remote Sens., vol. 189, pp. 220–235, 2022.

[6] S. Minaee et al., “Image segmentation using deep learning: A survey,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 7, pp. 3523–3542, Jul. 2022.

[7] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, “Zero-shot learning—A comprehensive evaluation of the good, the bad and the ugly,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, no. 9, pp. 2251–2265, Sep. 2019.

[8] T. Brandtberg and F. Walter, “Automated delineation of individual tree crowns in high spatial resolution aerial images by multiple-scale analysis,” Mach. Vis. Appl., vol. 11, pp. 64–73, 1998.

[9] F. E. Fassnacht et al., "Review of studies on tree species classification from remotely sensed data," Remote Sens. Environ., vol. 186, pp. 64–87, 2016.

[10] M. Freudenberg, P. Magdon, and N. Nölke, "Individual tree crown delineation in high-resolution remote sensing images based on U-Net," Neural Comput. Appl., vol. 34, pp. 22197–22207, 2022.

[11] J. Xu et al., "Tree crown segmentation and diameter at breast height prediction based on BlendMask in unmanned aerial vehicle imagery," Remote Sens., vol. 16, p. 368, 2024.

[12] S. Li et al., "Deep learning enables image-based tree counting, crown segmentation, and height prediction at national scale," PNAS Nexus, vol. 2, no. 4, p. pgad076, 2023.

[13] Y. Zhang et al., "Individual tree detection and counting based on high-resolution imagery and the canopy height model data," Geo-Spatial Inf. Sci., vol. 27, no. 6, pp. 2162–2178, 2024.

[14] M. Wielgosz et al., "SegmentAnyTree: A sensor and platform agnostic deep learning model for tree segmentation using laser scanning data," Remote Sens. Environ., vol. 313, p. 114367, 2024.

[15] M. S. R. Saimun and M. M. Rahman, "A comprehensive review of tree cover mapping using satellite sensor data," Discov. Geosci., vol. 3, p. 90, 2025.

[16] J. Holmgren, E. Lindberg, K. Olofsson, and H. J. Persson, "Tree crown segmentation in three dimensions using density models derived from airborne laser scanning," Int. J. Remote Sens., vol. 43, no. 1, pp. 299–329, 2022.

[17] M. Walker and G. Smith et al., "Literature review of unmanned aerial systems and LIDAR with application to distribution utility vegetation management," Arboric. Urban Forestry, vol. 49, no. 3, pp. 144–156, 2023.

[18] S. Yel and E. Gormus, "Exploiting hyperspectral and multispectral images in the detection of tree species: A review," Front. Remote Sens., vol. 4, 2023.

[19] F. Tong and Y. Zhang, "Individual tree crown delineation in high resolution aerial RGB imagery using StarDist-based model," Remote Sens. Environ., vol. 319, p. 114618, 2025.

[20] J. L. Vaschetti, E. Arnaudo, and C. Rossi, "TreePseCo: Scaling individual tree crown segmentation using large vision models," Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci., vol. XLVIII-M-7-2025, pp. 275–282, 2025.

[21] L. Yao, T. Liu, J. Qin, N. Lu, and C. Zhou, "Tree counting with high spatial-resolution satellite imagery based on deep neural networks," Ecol. Ind., vol. 125, p. 107591, 2021.

[22] A. A. Movassagh, J. A. Alzubi, M. Gheisari et al., "Artificial neural networks training algorithm integrating invasive weed optimization with differential evolutionary model," J. Ambient Intell. Human Comput., vol. 14, pp. 6017–6025, 2023.

[23] J. A. Alzubi, R. Jain, P. Nagrath et al., "Deep image captioning using an ensemble of CNN and LSTM based deep neural networks," J. Intell. Fuzzy Syst., vol. 40, no. 4, pp. 5761–5769, 2020.

[24] M. C. Aksoy, Trees in Satellite Imagery, Kaggle Dataset, 2021. Available: https://www.kaggle.com/datasets/mcagriaksoy/trees-in-satellite-imagery