scientific reports

Check for updates

OPEN

A UAV RGB dataset and method for instance tree crown segmentation for biodiversity monitoring

Mai Viet Hoang Do[1], Duc-Thang Phung[1], Hoang Duy Linh Pham[1], Quang-Duy Pham[1], Van-Nam Hoang[1], Van-Sam Hoang[2], Michiel Vlaminck[3], Hiep Luong[3], Thanh-Hai Tran[1], Hai Vu[1] & Thi-Lan Le[1]✉

The use of UAVs (Unmanned Aerial Vehicles) equipped with cameras has emerged as a promising approach due to its efficiency and accuracy in assessing biodiversity. Tree instance segmentation plays a crucial role in UAV image analysis, serving as a foundational step for subsequent tasks such as species identification. However, instance segmentation of trees from UAV images presents significant challenges, especially in dense forest environments. This leads to severe ambiguity in determining instance boundaries. Rather than directly segmenting entire tree crowns—which often fails under such challenging conditions—we adopt a deliberate over-segmentation strategy based on a contour detection network. Subsequently, the contours are merged to produce more accurate segmentation results. These ideas are integrated into the first contribution of the paper: TreeCoG - a method for instance-level tree crown segmentation from UAV-captured RGB images. TreeCoG consists of three main steps: (1) contour extraction, (2) contour feature extraction and representation, and (3) contour merging. The first step extracts contour candidates from UAV images, while the second and the third steps construct a graph from these contours and employ a Graph Convolutional Network (GCN) to merge them into instances. The second contribution of this paper is a new image dataset, ForestSeg, collected using UAVs in dense tropical forests of Vietnam in multiple seasons. The ForestSeg dataset is composed of four subsets—ForestSeg-T1, ForestSeg-T2, ForestSeg-T3, and ForestSeg-T4 comprising a total of 2,944 annotated images. These subsets were acquired at different time periods and flight altitudes, thereby capturing variations in tree appearance and supporting robust evaluation of instance tree crown segmentation methods. This dataset enables the analysis of temporal variations in tree characteristics and serves as a valuable resource for evaluating the robustness of segmentation methods under varying tree appearances over time. We conduct extensive experiments to assess the contribution of each component of TreeCoG and compare our approach with the state-of-the-art methods on two datasets: our self-collected dataset, ForestSeg, and the BAMFORESTS dataset. Experimental results demonstrate the superior performance of TreeCoG in tree instance segmentation, achieving 57.01% AP, 62.21% AP@50, and 55.32% AP@70 on the ForestSeg dataset, and 53.21% AP, 73.14% AP@50, and 43.23% AP@70 on the BAMFORESTS dataset, respectively. These results confirm the effectiveness of the proposed method in accurately delineating individual tree instances.

Keywords Remote sensing, Instance segmentation, Biodiversity monitoring

With the growing population and associated environmental challenges such as climate change, land degradation, and ecological imbalance, biodiversity loss has become a critical issue. Monitoring biodiversity, especially in trees, plays a crucial role in environmental protection[1]. Traditional methods that rely on manual monitoring are time-consuming and cannot provide continuous monitoring. The advancement of remote sensing devices has opened new possibilities for automated biodiversity monitoring[2]. Thanks to their flexibility in flying altitudes and sensor integration, Unmanned Aerial Vehicles (UAVs) are increasingly used for resource management and visual inspection tasks, such as Bark Beetle damage detection[3] and tree classification[4].

Tree instance segmentation is a crucial step in UAV image analysis for biodiversity monitoring[5]. It provides valuable information for subsequent tasks such as forest structure analysis, species identification, species

[1]School of Electrical and Electronic Engineering (SEEE), Hanoi University of Science and Technology, Ha Noi 100000, Vietnam. [2]Vietnam National Forestry University, Ha Noi 10000, Vietnam. [3]Image Processing and Interpretation (IPI), UAV Research Centre, Ghent University, IMEC, Ghent, Belgium. ✉email: lan.lethi1@hust.edu.vn

distribution mapping, and ecological health assessment[6]. However, instance segmentation methods from UAV images in biodiversity monitoring face substantial challenges, particularly when trees of the different species exhibit similar visual characteristics. These challenges stem from the inherent similarities in the shapes, sizes, and textures of trees belonging to the same species, which make it difficult for segmentation algorithms to accurately distinguish between closely spaced or overlapping trees. Furthermore, variations in environmental factors such as lighting, shadows, and the structural complexity of tree canopies further compound these difficulties. Consequently, current methods must adopt advanced approaches, including machine learning, deep learning, or multi-sensor fusion, to enhance segmentation accuracy and address these limitations effectively.

Our study aims at developing instance tree segmentation from UAV images for dense forest environments. In dense forest, the ambiguity in tree instance boundaries, severe crown overlap and occlusion and texture similarity can significantly degrade the segmentation accuracy of conventional methods. Rather than directly segmenting entire tree crowns—which often fails under such challenging conditions—we adopt a deliberate over-segmentation strategy based on a contour detection network. Subsequently, the contours are merged to produce more accurate segmentation results. These ideas are integrated into the first contribution of the paper: TreeCoG - a method for instance-level tree crown segmentation from UAV-captured RGB images that consists of three main steps: (1) contour extraction, (2) contour feature extraction and representation, and (3) contour merging. In the first step, a deep edge transformer (EDTER)[7] is employed to decompose the canopy into smaller and atomic contours. Then, shape and appearance features are then computed in the second step for contour representation and merging. Regarding the contour merging, we adopt the method based on Graph Neural Network (GNN) in[8]. Then, a new UAV image dataset named ForestSeg collected from dense tropical forests in Vietnam is the second contribution of the paper. The ForestSeg dataset is composed of four subsets–ForestSeg-T1, ForestSeg-T2, ForestSeg-T3, and ForestSeg-T4 comprising a total of 2,944 annotated images. The dataset is notable for spanning multiple seasons, enabling the analysis of temporal variations in tree characteristics. We conduct extensive experiments to assess the contribution of each component of TreeCoG and compare our approach with the state-of-the-art methods on two datasets: our self-collected dataset, ForestSeg, and the BAMFORESTS dataset. Experimental results demonstrate the superior performance of TreeCoG in tree instance segmentation, achieving 57.01% AP, 62.21% AP@50, and 55.32% AP@70 on the ForestSeg dataset, and 53.21% AP, 73.14% AP@50, and 43.23% AP@70 on the BAMFORESTS dataset, respectively. In summary, the contributions of this paper are twofold:

- TreeCoG: a method for instance-level tree crown segmentation from UAV-captured RGB images. By leveraging over-segmentation and contour-based merging strategies, TreeCoG achieves robust performance in dense forest environments.
- ForestSeg: a new UAV image dataset of dense tropical forests in Vietnam, captured across different time periods and flight altitudes, designed to support robust evaluation of instance tree crown segmentation methods.

By addressing the inherent challenges of instance segmentation in forestry applications, this study contributes to the advancement of automated tree analysis using UAV-based remote sensing, offering a solution for biodiversity monitoring.

This paper is organized into three main sections. First, we introduce our ForestSeg dataset and describe its key characteristics, including the challenges it presents for instance segmentation in densely vegetated environments. Second, we present our proposed method, detailing the comprehensive framework developed for accurate and efficient instance segmentation. Finally, experimental results are presented to demonstrate the effectiveness of our approach.

Related works

Application of remote sensing for forest and biodiversity monitoring

Forest and biodiversity monitoring increasingly rely on computer vision techniques to automatically analyze large-scale remote sensing data. Among these techniques, detection, segmentation, and tracking constitute the core tasks. Detection aims to localize objects of interest, such as individual trees, while segmentation provides fine-grained, pixel-level delineation of vegetation and tree crowns. Tracking further extends these tasks by associating detected or segmented instances across time, enabling the analysis of forest dynamics, tree growth, and ecosystem changes as well as object movement and behavior[9,10]. Together, these tasks support quantitative assessment of forest structure, health, and biodiversity from UAV and remote sensing imagery.

Regarding detection methods, in UAV and remote sensing imagery, majority of methods focus on tree crown detection and delineation (ITCD). According to a comprehensive survey by Zheng et al., ITCD methods can be categorized into three main groups: traditional image processing, traditional machine learning, and deep learning approaches[11]. While traditional methods often struggle in dense or overlapping forest areas, deep learning methods particularly those based on object detection, have become the predominant trend due to their robust feature extraction capabilities and high accuracy on high-resolution data[11]. However, applying deep learning to practical scenarios still faces two major challenges: the need for detailed tree status monitoring and data discrepancies across geographical regions. Regarding the challenge of detailed monitoring, most previous studies have focused solely on tree counting or binary classification. To address this, Zheng et al. (2021) proposed the MOPAD model using UAV images to not only detect oil palm trees but also classify them into five growing statuses (e.g., dead, yellowish, or mismanaged trees)[12]. This study effectively resolved the issue of extreme class imbalance among tree categories through a Refined Pyramid Feature (RPF) module and a hybrid class-balanced loss function[12]. Regarding the challenge of geographical scalability, variations in sensors and environmental conditions often lead to performance degradation. To overcome this, Zheng et al. (2020) developed a Multi-level Attention Domain Adaptation Network (MADAN) for cross-regional oil palm tree counting[13]. By utilizing a

Multi-level Attention mechanism and Batch-Instance Normalization (BIN), this method allows for the effective transfer of knowledge from labeled source domains to unlabeled target domains[13]. Going further to address the limitations of domain adaptation, Zheng et al. proposed a Domain Generalization approach with the MMD-DRCN model[14]. By combining data from multiple sources and employing Maximum Mean Discrepancy (MMD) loss to align feature distributions, this method can accurately detect trees in completely unseen target domains without the need for prior data collection, opening new directions for global-scale monitoring[14]. Furthermore, some methods that improves U-shape architectures[15,16] for object detection could be applied for forest UAV images.

Image segmentation is critical for analyzing very high-resolution remote sensing imagery, yet traditional superpixel methods often fail to delineate complete geo-objects due to intra-object heterogeneity and inter-object homogeneity. While Region Adjacency Graph (RAG) based merging is a standard solution, it typically relies on handcrafted features and necessitates the manual, trial-and-error selection of scale parameters. To address these limitations, Lv et al. (2025) proposed DeepMerge, a weakly supervised framework that integrates deep learning with RAGs[17]. Instead of requiring pixel-wise object labels, DeepMerge learns similarity metrics from adjacent superpixel pairs using a Siamese network with a cross-scale transformer backbone[17].

Recently, large pre-trained vision models such as the Segment Anything Model (SAM), known for their strong generalization and zero-shot capabilities, have been widely adopted for object segmentation across various domains[18]. In[19], SAM-based methods were applied to automatically segment individual tree crowns from UAV imagery of young plantations. However, the authors observed that SAM, even when guided with carefully designed prompts, did not outperform a customized Mask R-CNN. To address this limitation, they proposed DSMPrompter, a new method that integrates Digital Surface Model (DSM) information into the RSPrompter framework[20].

Besides tree crown detection and segmentation, recent studies have also focused on forest health classification[21]. This task aims to assess the condition of forests or individual trees and to identify signs of stress, disease, or degradation. To address this problem, a variety of techniques[22,23] could be applied.

Instance segmentation and tree instance segmentation

Instance segmentation tasks from UAV-captured images play a crucial role in accurately delineating individual objects within complex natural environments. Mask R-CNN[24] is a leading deep learning framework that combines Region Proposal Networks (RPN) with mask prediction, thereby enabling simultaneous object detection and pixel-level segmentation. This characteristic makes it particularly well-suited for processing UAV RGB imagery. For example, the authors in[25] applied Mask R-CNN[24] for the segmentation of olive tree crowns and shadows, achieving F1 scores that exceeded 94% at a resolution of 3 cm/pixel. Their study demonstrated not only the method's high accuracy in biovolume estimation but also its robust capability to handle irregular tree shapes, as evidenced by strong precision and recall value across different spectral bands. Furthermore, in[26], the authors introduced a modified version of Mask R-CNN[24] for individual tree segmentation task, referred to as MCAN, which integrates a Cross Stage Partial Net (CSP-Net) along with a Convolutional Block Attention Module (CBAM). This enhanced architecture significantly improved detection accuracy in urban forest segmentation, particularly in complex landscapes, by bolstering feature extraction and facilitating effective multiscale segmentation.

The YOLO (You Only Look Once) family, particularly YOLOv5 and its subsequent variants, has been widely adapted for instance segmentation tasks, offering the real-time processing capabilities crucial for large-scale UAV surveys[27]. These models predict both bounding boxes and segmentation masks simultaneously, thereby balancing speed and accuracy. For example, the authors in[28] explored the use of YOLOv5 for instance segmentation of individual tree crowns on the ForInstance benchmark LiDAR dataset, achieving competitive results across diverse forest types. Their findings highlight that while YOLO-based methods deliver processing speeds suitable for real-time applications, performance may vary with canopy density. Reported mAP50 scores—such as a value of 0.902 in applications like weed detection in potato fields, which can be extended to tree segmentation—underscore YOLO's potential. Further, adjustments in hyperparameters, such as those implemented by the Task-Aligned Assigner in YOLOv8E for Trees, have demonstrated additional improvements in canopy segmentation accuracy[29]. Despite these advantages, YOLO's lightweight design[30], which is ideal for resource-constrained environments, sometimes struggles to capture the fine-grained details found in dense forest scenes.

U-Net and its related approaches have long been a cornerstone in semantic segmentation due to their effective encoder-decoder architectures that capture both local and global contextual information[31]. Although U-Net is primarily tailored for semantic segmentation, recent adaptations have sought to extend its capabilities to instance segmentation by incorporating clustering or detection methods. In the domain of tree segmentation, the authors in[32] illustrates the utility of U-Net as part of a pipeline for tree crown segmentation in simpler forest stands. In these applications, U-Net serves as a preliminary segmentation step to isolate potential tree crowns, which are then refined through additional post-processing to resolve overlapping instances. Performance metrics from several studies indicate that U-Net-based methods can achieve high semantic segmentation accuracy. However, their effectiveness at the instance level is less well-documented, with persistent challenges in accurately distinguishing overlapping trees. These limitations underscore that, while U-Net provides a robust foundation for extracting salient features from UAV-acquired imagery, its direct application for instance segmentation requires significant adaptation.

Several methods have been developed to enhance the performance of tree instance segmentation. One notable approach focuses on edge detection in trees. In[33], the authors introduced a novel instance segmentation method for high-resolution remote sensing images, named QuadTransPointRend (QTPR-Net). This method combines the PEFE module and the EPFR module. First, the PEFE module is employed for edge feature extraction, progressively adding coarse-grained features layer by layer. After multi-level feature fusion, uncertain points are identified. Next, the EPFR module, utilizing a transformer-based structure, captures diverse contextual information to generate precise mask predictions for edge pixels. This architecture was evaluated on three different datasets, demonstrating the effectiveness of the quadtree attention mechanism in edge segmentation. By employing a coarse-to-fine pyramid approach, it enhances attention to uncertain points and incorporates multi-scale positional encoding to improve efficiency and reduce loss. Additionally, in[34] the authors proposed a method for performing the delineation of individual tree crowns named Detectree2 by adapting Mask-RCNN[24] with ResNet101 backbone. They applied their best models for each site to their entire tiled orthomosaics to generate site wide crown maps. They combined these crown delineations with repeat lidar surveys to determine the height changes in individual trees in their four sites.

Moreover, in[8], the authors employed a graph-based approach to analyze tree regions. They utilized a deep edge detection method[35] to initiate contour detection, followed by the extraction of spatial and temporal features. To process these features, a Graph Neural Network (GNN) was employed to model the relationships between adjacent contours. Inspired by this work, our study also leverages contour-based analysis and Graph Neural Network models. However, instead of incorporating both spatial and temporal information, we focus solely on extracting spatial features to enhance tree instance segmentation.

UAV-based forestry datasets

The increasing availability of high-resolution imagery captured by Unmanned Aerial Vehicles has opened new opportunities for precise environmental monitoring and forestry management. In this context, tree instance segmentation, which aims to delineate and distinguish individual trees within dense forest canopies, has emerged as a critical task for application such as biomass estimation, species identification, and ecosystem health assessment. Recent advances in UAV-based remote sensing have led to the development of a diverse range of forestry datasets that collectively address various challenges in forest monitoring and management. For instance, the UAVS-FFDB[36] dataset focuses on forest fire detection by providing high-resolution aerial imagery along with detailed annotations categorizing pre-fire and post-fire conditions, thereby supporting early-warning systems and real-time damage assessments. In contrast, the BAMFORESTS[37] dataset offers extremely high-resolution UAV imagery with individual tree crown delineations that enable precise tree segmentation and biomass estimation, proving invaluable for instance segmentation tasks in dense forest canopies.

Furthermore, the FOR-instance[38] dataset contributes a benchmark of UAV laser scanning data, manually annotated for both semantic and instance segmentation of individual trees, which is particularly beneficial for developing and benchmarking 3D segmentation algorithms as well as for deriving critical forest inventory parameters such as diameter at breast height (DBH). Expanding on LiDAR-based approaches, TreeScope[39] offers a dataset that leverages LiDAR data collected from both UAV and mobile robotic platforms to support semantic segmentation and precise mapping of tree stems, facilitating applications in precision forestry by enabling accurate DBH measurements.

In tropical forestry, ReforesTree[40] uniquely combines low-cost RGB UAV imagery with deep learning methodologies to estimate forest carbon stock, thereby addressing the pressing need for scalable carbon monitoring in agro-forestry systems. Additionally, the Open Forest Observatory aggregates over 150 drone-based datasets—including raw imagery, orthomosaics, canopy height models, and ground inventory data—to provide a comprehensive platform for calibrating and validating UAV-based remote sensing workflows across diverse forest ecosystems.

Finally, a dataset designed for Task Planning Support for Arborists and Foresters integrates UAV-derived RGB[41] and multispectral imagery with supplementary satellite and soil moisture data, thereby facilitating detailed tree inventory and vitality assessments that are essential for effective urban and forest management planning. Collectively, these datasets represent a robust suite of resources that not only advance the state-of-the-art in UAV-based forestry monitoring but also highlight the importance of integrating multiple data modalities to address the multifaceted challenges of modern forest management.

ForestSeg: a new dataset for instance tree segmentation

Study area

This study was conducted at the Vietnam National University of Forestry (VNUF), situated in Xuan Mai town, Chuong My district, Hanoi, Vietnam. The surrounding forest environment was selected as the research site due to its rich biodiversity, complex terrain, and dense vegetation, which together provide a realistic and challenging setting for evaluating UAV-based aerial survey methodologies. The forested area spans approximately 110 hectares, encompassing a variety of vegetation types, including natural dense forests, systematically planted tree stands, and experimental plots dedicated to scientific research and conservation. These diverse forest compositions offer a valuable testbed for assessing UAV performance in different ecological conditions.

Within this larger forest landscape, a 25-hectare study site was strategically chosen for data collection. The selection criteria included terrain variability, canopy density, and site accessibility, ensuring that the chosen area would present a representative set of challenges commonly faced in UAV-based forest monitoring. This diversity of environmental conditions allowed for a thorough evaluation of the proposed instance segmentation approach across different forest structures and canopy complexities.

UAV platform

The dataset includes high-resolution UAV imagery collected at multiple altitudes (70–211 m) across various dates and seasons. Flight missions produced between 134–745 images, depending on flight altitude, overlap, and duration. Two UAV models were used:

Dataset name	Date	Time	Altitude	Original images	Resolution	Train	Test	Area
ForestSeg-T1	22/5/2024	14:40–14:50	70m	134	5472 × 3648 px	1344	480	7.1 ha
ForestSeg-T2	10/8/2024	16:50–17:15	211m	234	8064 × 6048 px	–	410	19.3 ha
ForestSeg-T3	24/8/2024	12:30–13:30	150m	720	8064 × 6048 px	–	350	24.5 ha
ForestSeg-T4	24/8/2024	14:10–16:00	100m	745	8064 × 6048 px	–	360	12.5 ha

Table 1. The information of four sets in ForestSeg dataset.

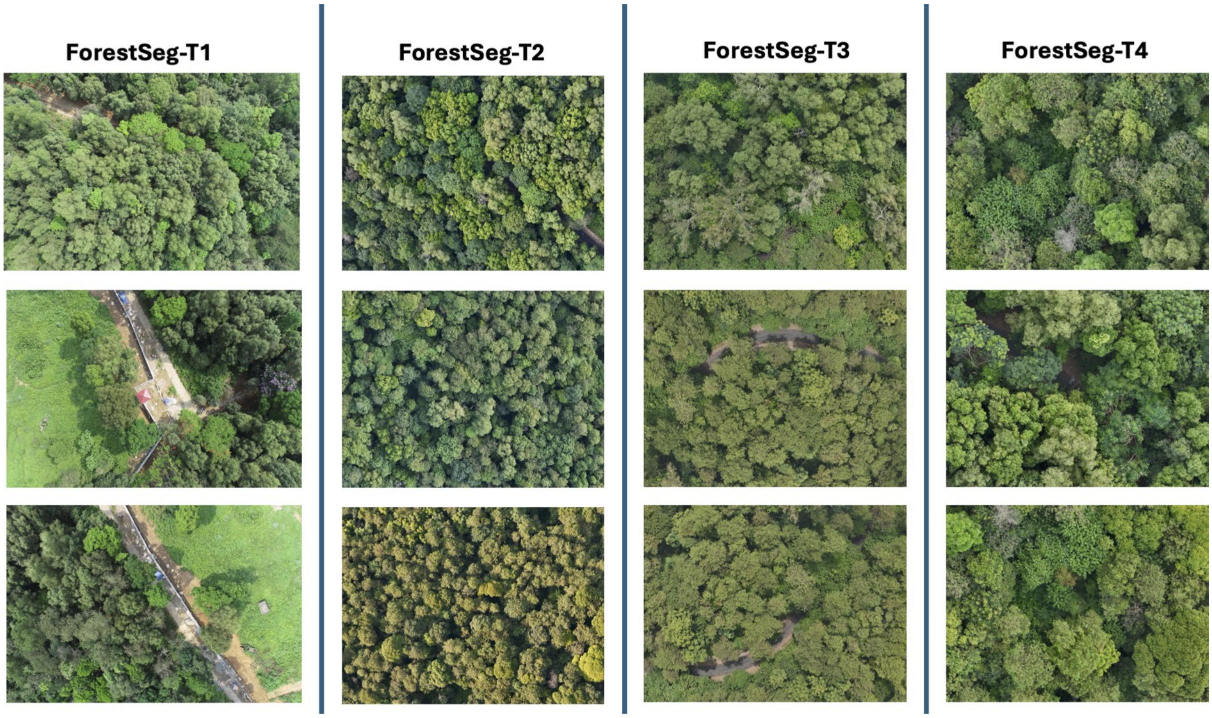

Fig. 1. Some samples in four sets of ForestSeg dataset described in Table 1.

- DJI Air 3: imagery at 8064 × 6048 px
- DJI Phantom 4 RTK: imagery at 5472 × 3648 px

Four flights have been conducted. Each flight includes metadata on altitude, resolution, coverage area, and the number of captured images (see Table 1). The multi-altitude, multi-season nature of the dataset supports forest monitoring, biomass estimation, land cover classification, and environmental change detection. High-resolution imagery allows for detailed feature extraction, such as individual tree delineation, canopy density analysis, and terrain modeling.

To ensure geospatial accuracy, Ground Control Points (GCPs) were strategically distributed and measured using RTK-GNS receivers, achieving centimeter-level precision. The GCP layout accounted for variations in topography and canopy density, minimizing distortion during orthorectification and 3D reconstruction.

Dataset

We performed manual annotation, relying on human experts to mark tree crowns based on visual cues like texture gradients, color variation, and canopy structure. Images are selected from four flights to do annotation.

To facilitate efficient processing and training the segmentation models, the images were divided into smaller patches of 1024 × 1024 pixels with the overlap ratio of 60% (Fig. 1).

Our dataset, ForestSeg, comprises four subsets corresponding to four different drone flights: ForestSeg-T1, ForestSeg-T2, ForestSeg-T3, and ForestSeg-T4. ForestSeg-T1 includes a total of 1,824 annotated images, of which 1,344 are used for training and 480 for evaluation. To support model assessment under varying conditions and across different time intervals, we additionally annotated three test sets: ForestSeg-T2, ForestSeg-T3, and ForestSeg-T4, containing 410, 350, and 360 images, respectively. In this study, these three sets are used exclusively for cross-season/cross-time evaluation. This means the models trained on the ForestSeg-T1's training set will be evaluated on ForestSeg-T2, ForestSeg-T3, and ForestSeg-T4. Detailed information about each set is provided in Table 1. Figure 1 illustrates some samples in ForestSeg dataset. The distribution of tree instances per image in ForestSeg-T1 is shown in Fig. 2. We can observe that the number of instances ranges from 1 to 17. Most images

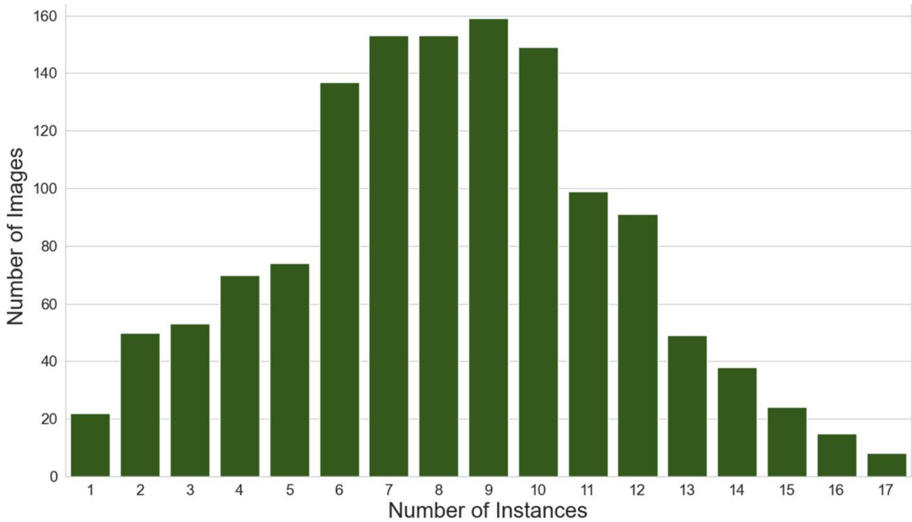

Fig. 2. Distribution of images by the number of tree instances in the ForestSeg-T1 dataset.

Dataset	Sensor modality	Geographic region	Temporal coverage	Annotation completeness	Instances	Resolution
BAMFORESTS[37]	UAV RGB imagery	Bamberg, Germany	Single season	Focus on tree crowns; understory not labeled	∼27,160 tree crowns	∼1.6–1.8 cm GSD (orthomosaics)
SiDroForest	UAV RGB imagery, SfM point clouds	Chukotka & Yakutia, Siberia	Mainly single acquisition	High for trees >10 m; smaller trees less consistent	∼19,342 detected crowns	Sub-decimeter (few cm)
FOR-instance[38]	UAV LiDAR	Multiple (Norway, Czech Rep., Austria, NZ, Australia)	Single acquisition per dataset	Complete segmentation of trees & background	>1,000 aggregated	High-density LiDAR (fine 3D details)
TreeScope[39]	UAV & mobile robotics LiDAR	Agricultural environments	Single time point	High quality for orchards	>1,800 annotated stems	Millimeter-level precision
ReforesTree[40]	UAV RGB imagery	Tropical forests (Ecuador)	Limited (primarily one season)	Complete for individual crowns	Numerous (exact count not specified)	Ultra-high (few cm GSD)
Our Dataset	UAV RGB	Tropical forest (Viet Nam)	**Multi-season** (data from different seasons)	**Very high; complete labeling of trees**	∼2000 tree crowns	**Ultra-high (3 cm GSD)**

Table 2. Comparison of UAV-based forestry datasets for instance segmentation.

contain between 6 and 10 trees, with notable peaks at 8 and 9 each appearing in approximately 160 images suggesting a moderate canopy density.

Table 2 presents a comparison with existing UAV-based forestry datasets. While datasets such as BAMFORESTS[37] and FOR-instance[38] offer high-resolution annotations and LiDAR-derived structure, they often lack in temporal coverage or general applicability. UAV Tree Identification and ReforesTree[40] contribute valuable ecological features but are limited in geographic and season. In contrast, our dataset provides high-resolution RGB images captured from UAV in multi-seasons at multi-altitude coverages. In addition, the studied forest area has rich canopy structure. These features make it a comprehensive benchmark for instance segmentation and long-term forest monitoring applications. The dataset is publicly available at website [https://sigm-seee.github.io/datasets/ForestSeg.html].

Although the ForestSeg dataset provides valuable information for training and evaluating segmentation models, its annotation quality still presents certain limitations. First, manual annotations are often influenced by subjectivity and the annotator's experience, which can lead to boundary inconsistencies, especially for small or overlapping objects. Second, due to the complexity of large-scale imagery, some instances may be partially mislabeled or omitted, reducing annotation completeness. Finally, the annotation process is costly and time-consuming, which restricts the dataset's scale and diversity. These limitations may introduce biases into the model training process and affect the generalizability of the results.

TreeCoG: the proposed method for tree crown segmentation

Although several instance segmentation methods have been proposed[42], the dense foliage and complex structure of tree crowns make segmentation particularly challenging. In some cases, even humans may struggle to determine whether a region belongs to the same tree crown. Therefore, in this paper, we formulate tree crown segmentation as a graph construction and learning problem. For a given image, a graph is initialized based on the results of an edge detection method. Each node in the graph represents a tree canopy contour, while the weight of the graph edge between nodes is determined by the similarity between them. Due to the complexity of tree canopies and the high similarity among them, regions belonging to the same tree crown may be divided into several contours (i.e., nodes). To address this, a learning process is employed to determine whether these nodes should be merged. Figure 3 illustrates the overall framework of the proposed method which involves three stages: contour extraction, contour features construction and representation, and contour merging. The result of this process is output masks of segmented tree regions.

Contour extraction from images

As we have mentioned, in dense forest scenes, individual tree crowns frequently merge visually from UAV views. This leads to the severe ambiguity in determining instance boundaries. Instead of attempting to directly segment entire tree crowns which is an approach that often fails under these conditions, we adopt a deliberate over-segmentation strategy based on contour detection network. Because the characteristics of Deep Edge Network[43], this steps decomposes the canopy into smaller and atomic regions that are unlikely to span multiple crowns.

For the contour initiation task, we first applied a deep edge transformer (EDTER)[7]. This method was introduced to capture both coarse-grained global context and fine-grained local context through a two-stage process. Additionally, it introduces a novel Bi-directional Multi-Level Aggregation (BiMLA) decoder that is particularly well-suited for remote sensing imagery because it effectively fuses multi-scale features while preserving the fine spatial details essential in such data. BiMLA has dual-path design, where the top-down path propagates high-level semantic information and the bottom-up path reinforces low-level spatial cues. Additionally, its learnable upsampling mechanism adapts the reconstruction of high-resolution details to the inherent noise and variability found in satellite and aerial imagery. This robust integration of global context and local precision not only enhances edge detection performance but also leads to cleaner, more reliable representations. As extracted edges often contain a considerable amount of noise, which can negatively impact subsequent processing steps, we apply the Gaussian Blur method to mitigate this issue, which smooths the image and reduces high-frequency noise, thereby improving the quality of the edge map. At the same time, we employ the Guo-Hall algorithm[44] to skeletonize the edge map, ensuring that the extracted edges are represented as thin, well-defined structures. This skeletonization process helps preserve the essential topological properties of the edges while eliminating redundant pixels. Once these steps are completed, a refined edge map is obtained, which serves as a crucial input for constructing the graph in the subsequent steps. Figure 4 illustrates the output of contour extraction step.

While this may initially fragment a single tree, it greatly reduces the chance of incorrectly merging multiple adjacent crowns which is a critical concern in high density regions.

Feature extraction and contour representation

Once the contours are generated in the previous step, a graph is constructed where each node represents a contour and each edge encodes the relationship between a pair of contours. For each node, a five-dimensional shape feature vector is computed, while appearance features are used to calculate the weights of the graph edges. The weight of the graph edge is defined by the appearance similarity of two patches defined by the correspond contours.

Appearance features and contour similarity score

To compute the similarity score between two patches, Learned Perceptual Image Patch Similarity (LPIPS) metric[45] is employed as it aligns closely with human perceptual judgments of image similarity. For each contour, $p \times p$ RGB image patch is extracted. Then, a convolutional neural network based on AlexNet is used to extract deep feature representations from the input patches. Finally, we compute the cosine similarity between the feature vectors. The weights of edge graph are then defined by the cosine similarity. Figure 5 illustrates the similarity score computed for two contours. In detail, this process is formulated as follows:

Given two contours C_i and C_j, we first extract the corresponding RGB image patches $P_i, P_j \in \mathbb{R}^{p\times p\times 3}$. Each patch is then fed into an AlexNet-based convolutional neural network $\Phi(\cdot)$ to obtain deep feature representations:

$$\mathbf{f}_i = \Phi(P_i), \quad \mathbf{f}_j = \Phi(P_j), \tag{1}$$

where $\mathbf{f}_i, \mathbf{f}_j \in \mathbb{R}^d$ denote the d-dimensional feature vectors.

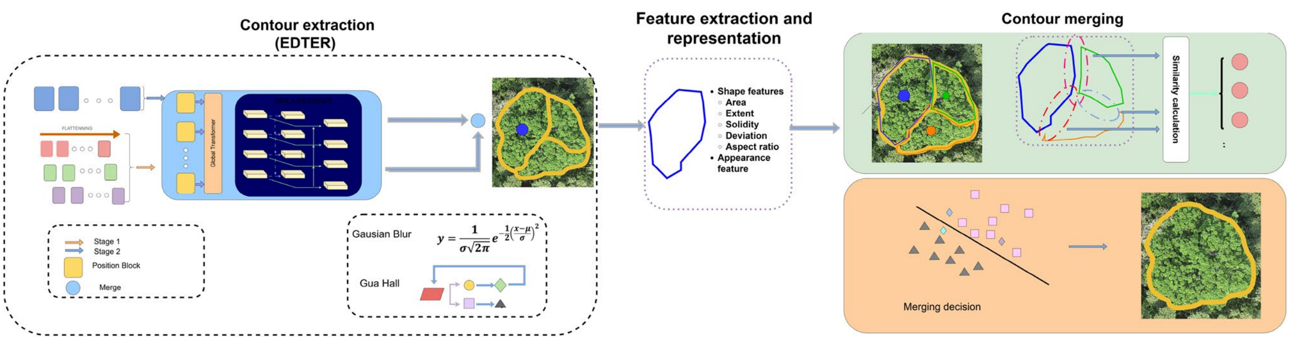

Fig. 3. The pipeline of proposed method for tree instance segmentation comprising of three main steps: contour extraction, contour feature extraction and representation, and contour merging.

Fig. 4. Visualization of the results following the contour extraction step in ForestSeg-T1 dataset.

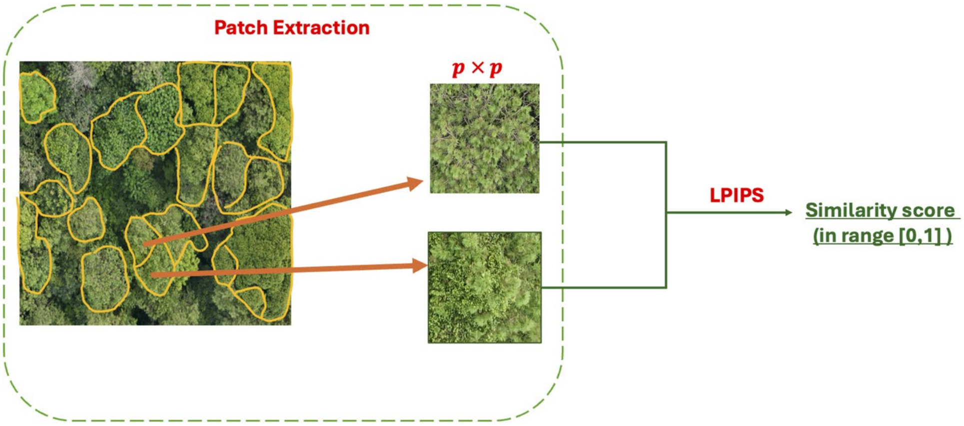

Fig. 5. Similarity score of two patches computed from LPIPS[45].

The appearance-based similarity between the two contours is computed using the cosine similarity:

$$s_{ij} = \cos(\mathbf{f}_i, \mathbf{f}_j) = \frac{\mathbf{f}_i^\top \mathbf{f}_j}{\|\mathbf{f}_i\|_2 \, \|\mathbf{f}_j\|_2}. \tag{2}$$

Finally, the edge weight of the graph is defined as:

$$w_{ij} = s_{ij}, \quad (i, j) \in \mathscr{E}, \tag{3}$$

where $\mathscr{E}$ denotes the set of edges in the graph.

By incorporating these weights, the graph effectively captures the underlying structure of the image, emphasizing meaningful connections while de-emphasizing less relevant ones. In the absence of these similarity-based weights, the input graph becomes an unweighted graph, where only the connectivity between contours is considered, without accounting for the strength of their relationships.

Shape features

In this study, five shape features–area, extent, solidity, aspect ratio, and deviation–are computed to represent the shape of a contour. These descriptors capture essential geometric properties, enabling more precise differentiation between contour structures. Figure 6 illustrates these shape features.

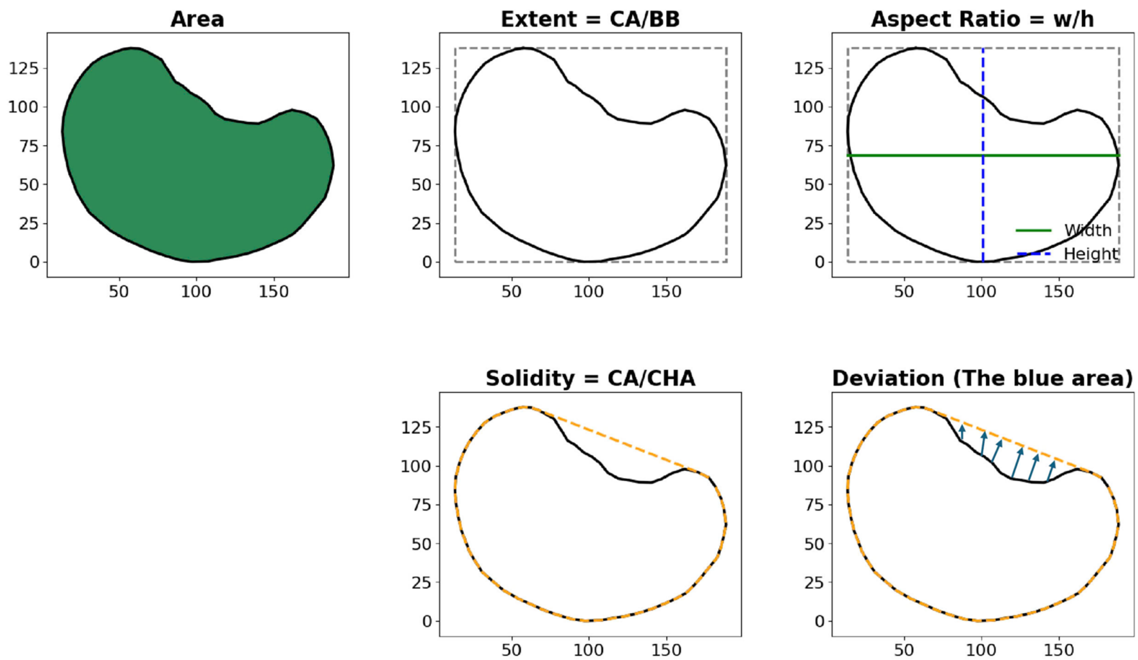

Fig. 6. Shape features extracted for contours.

Area represents the total number of pixels enclosed by the contour, quantifying the physical size of the tree crown. Larger crowns typically correspond to mature trees, while smaller areas may indicate young or underrepresented canopies. This measure provides a fundamental scale reference for distinguishing trees across heterogeneous forest stands:

$$A = \sum_{(x,y)\in\mathbb{Z}^2} \mathbf{1}_C(x,y) \tag{4}$$

where *A* is the total number of pixels enclosed by the contour *C*.

Extent measures the ratio of the contour area to the area of its bounding box, capturing how efficiently the crown occupies its minimum enclosing rectangular region. High extent values indicate compact and well-contained canopies, while low values suggest elongated, sparse, or partially undersegmented crowns:

$$E = \frac{CA}{BB} \tag{5}$$

where: *CA* is the contour area (i.e., number of enclosed pixels). *BB* is the area of the bounding box, calculated as width $\times$ height of the minimum enclosing rectangle.

Solidity is the ratio of the contour area to the area of its convex hull. This measures how fully the contour fills its convex shape, useful for identifying fragmented or irregular canopies. High solidity indicates cohesive and dense canopies, while low solidity suggests crowns with gaps, indentations, or highly branched outlines. This property is effective for detecting undersegmented regions or naturally sparse species, providing insight into canopy completeness and structural complexity:

$$S = \frac{CA}{CHA} \tag{6}$$

where *CHA* is the convex hull area–i.e., the area of the smallest convex polygon that fully contains the contour.

Deviation evaluates the irregularity of the contour shape by comparing it with its convex hull. It reflects how much the contour deviates from a smooth or compact shape:

$$D = \frac{1}{L}\int_0^L \left\|p(s) - p^h(s)\right\| \, ds \tag{7}$$

where: L is the total arc length of the contour, $p(s)$ is a point on the contour parameterized by arc length s, $p^h(s)$ is the corresponding point on the convex hull, $\left\|p(s) - p^h(s)\right\|$ is the Euclidean distance between the contour point and its convex hull counterpart.

Aspect ratio characterizes the overall shape of the canopy–whether it is circular, elliptical, or elongated:

$$AR = \frac{w}{h} \tag{8}$$

where w and h are the width and height of the bounding box, respectively.

Graph contour merging

Due to the fragmented structure of canopy boundaries, a single tree crown may be split into multiple contours, which can result in inaccurate segmentation and over-segmentation errors. To address this issue, adjacent contours that belong to the same crown need to be merged. The definition of adjacent contours are formulated as follow:

Let C_i denote a contour and $d(C_i, C_j)$ the distance between two contours C_i and C_j (Euclidean boundary distance). We define the adjacency set of C_i as the K nearest contours according to this distance

$$\mathscr{N}_K(C_i) = \arg\min_{S \subseteq \{j \neq i\},\, |S|=K} \sum_{j \in S} d(C_i, C_j). \tag{9}$$

In other words, for each contour C_i, we sort all distances $\{d(C_i, C_j)\}_{j \neq i}$ in ascending order and select the top-K nearest neighbors.

Based on the distribution of tree instances shown in Fig.2, most images contain between 7 and 10 individual trees, with a peak at 9 instances. To maintain a consistent neighborhood size that reflects the typical crown density within an image, we set the number of adjacent contours K to 9, corresponding to the mode of the ground-truth instance distribution. This choice ensures that each contour is connected to a representative set of nearby contours while avoiding excessive graph connectivity in sparse scenes. We found that the number of generated contours is typically higher than the number of true tree crowns due to over-segmentation, we intentionally base K on the ground-truth prior rather than the contour-level distribution. The ground-truth statistics provide a reliable estimate of the natural spatial density of crowns, whereas the contour distribution after segmentation can vary significantly and be affected by noise or fragmentation artifacts. Using a fixed K derived from the ground-truth prior thus provides a stable and interpretable graph structure across samples, while the subsequent edge classification module is responsible for filtering out false adjacency links introduced by over-segmentation.

An example of this merging process is illustrated in Fig.7. In this work, we formulate contour merging as an edge classification problem on a graph, where each node corresponds to a contour and edges represent adjacency relationships between contours. The objective is to predict whether an edge indicates that two contours should be merged (i.e., they belong to the same tree crown) or kept separate (i.e., they correspond to different crowns).

We formulate $\mathscr{S}$ be the set of contours, a binary matrix is defined as follows:

$$\mathbf{Z} \in 0, 1^{|\mathscr{S}| \times |\mathscr{S}|} \tag{10}$$

where $\mathbf{Z}_{ab} = 1$ indicates that contour S_a and contour S_b should be merged into the same instance. Given a contour graph as input, our goal is to learn a model that predicts this merge indicator matrix. We cast the problem as one of edge classification on the graph defined over the contours.

First, we extract shape features for each node in the graph (see Sec. 4.2.2) and stack them into a node feature matrix:

$$\mathbf{X}_0 \in \mathbb{R}^{|\mathscr{S}| \times d} \tag{11}$$

Next, we encode the relationships between nodes with an adjacency matrix:

$$\mathscr{B} \in [0, 1]^{|\mathscr{S}| \times |\mathscr{S}|} \tag{12}$$

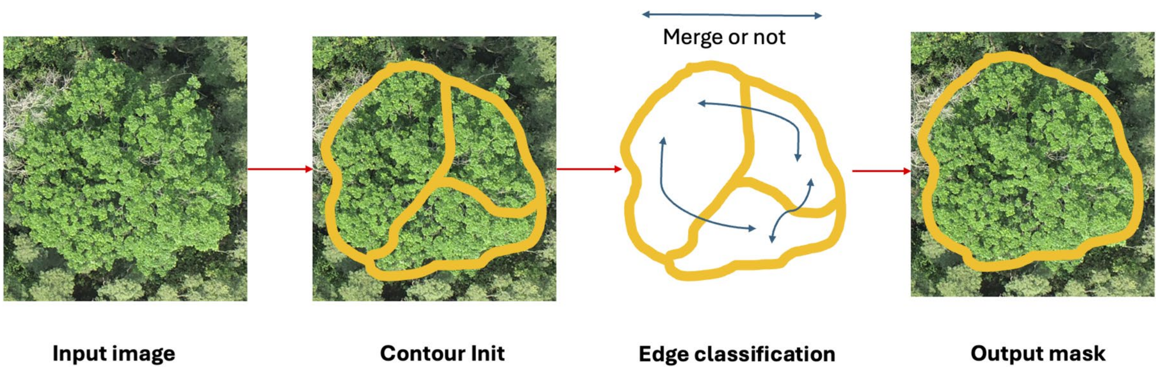

Fig. 7. Contour merging step based on GCN.

Using a graph convolutional layer (see, e.g.,[46]), we perform T rounds of message passing to aggregate information from each node's neighborhood:

$$\mathbf{X}^{(t+1)} = \phi\Big(\hat{\mathscr{B}}\,\mathbf{X}^{(t)}\,\Omega^{(t)}\Big), \tag{13}$$

for $t = 0, 1, \ldots, T-1$. Here, $\hat{\mathscr{B}}$ is the degree-normalized version of the adjacency matrix, ϕ is an element-wise non-linear activation function, and $\Omega^{(t)}$ are the trainable weights at the t^{th} layer.

To form the edge features for classification, we compute, following Mikolov et al.[47] the average of the final node features for each connected pair of nodes (S_a, S_b):

$$\mathbf{e}_{ab} = \frac{1}{2}\Big(\mathbf{X}_a^{(T)} + \mathbf{X}_b^{(T)}\Big) \tag{14}$$

Given the edge feature $\mathbf{e}_{ab}$, we then predict whether the corresponding contours should be merged by applying a multi-layer perceptron (MLP):

$$\hat{\mathbf{Z}}_{ab} = \mathscr{F}\big(\mathbf{e}_{ab}\big), \quad \forall (S_a, S_b) \text{ with } \mathscr{B}_{ab} = 1. \tag{15}$$

To train the model, we minimize the binary cross-entropy loss between the predicted merge matrix $\hat{\mathbf{Z}}$ and the ground truth matrix $\mathbf{Z}$, summing over all edges:

$$\mathscr{L} = -\sum_{\{(a,b):\mathscr{B}_{ab}=1\}} \Big[\mathbf{Z}_{ab}\log\big(\hat{\mathbf{Z}}_{ab}\big) + \big(1-\mathbf{Z}_{ab}\big)\log\Big(1-\hat{\mathbf{Z}}_{ab}\Big)\Big] \tag{16}$$

Finally, we merge the contours based on the predicted merge matrix $\hat{\mathbf{Z}}$ and produce an instance mask for each merged group. Specifically, for an instance represented as a tree τ, the predicted mask is given by the union of all pixels within the merged contours:

$$\mathscr{P}_\tau = \bigcup_{b:\,\mathbf{Z}_{\tau b}=1} S_b \tag{17}$$

Contour merged ground truth generation

The proposed procedure generates training labels for the contour merging stage by aligning extracted contours with instance-level ground truth annotations. First, contours are obtained from the edge detection mask. Each contour is then mapped to the most likely ground-truth instance by majority voting over the pixels it covers. Using this mapping, a binary merge matrix is constructed to indicate whether two contours should be merged (i.e., they belong to the same instance). Finally, a contour adjacency graph is built, and each edge is assigned a label derived from the merge matrix. This ensures that the graph representation provides consistent supervision for learning contour merging, where positive edges correspond to parts of the same object and negative edges correspond to different instances. The detailed implementation of this process is illustrated in Algorithm 1.

Require: Edge detection mask $E \in \{0, 255\}^{H \times W}$
Require: Instance segmentation ground truth $I_{GT} \in \mathbb{N}^{H \times W}$
Ensure: Merge matrix $M \in \{0, 1\}^{N \times N}$, where N is the number of contours

Step 1: Extract Contours
$\mathscr{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_N\} \leftarrow \text{ExtractContours}(E)$
$N \leftarrow |\mathscr{C}|$ {Number of contours}

Step 2: Map Contours to Instances
Initialize $\phi : \{1, \ldots, N\} \rightarrow \mathbb{N}$ {Contour-to-instance mapping}
for $i = 1$ **to** N **do**
    $\text{mask}_i \leftarrow \text{RenderContour}(\mathbf{c}_i)$
    $\mathscr{P}_i \leftarrow \{(y, x) \mid \text{mask}_i[y, x] > 0\}$
    $\text{instances} \leftarrow \{I_{GT}[y, x] \mid (y, x) \in \mathscr{P}_i \wedge I_{GT}[y, x] > 0\}$
    $\phi(i) \leftarrow \arg\max_{id} \text{Count}(id \text{ in instances})$
end for

Step 3: Build Merge Matrix
Initialize $M \in \{0, 1\}^{N \times N}$ with zeros
for $i = 1$ **to** N **do**
    for $j = i + 1$ **to** N **do**
        if $\phi(i) = \phi(j)$ **then**
            $M[i, j] \leftarrow 1$
            $M[j, i] \leftarrow 1$
        end if
    end for
end for

Step 4: Generate Edge Labels for Graph
$G = (V, \mathscr{E}) \leftarrow \text{BuildContourGraph}(\mathscr{C})$ $\{V = \{1, \ldots, N\}\}$
Initialize $L = []$
for each edge $(i, j) \in \mathscr{E}$ **do**
    $L.\text{append}(M[i, j])$
end for
return M, L, ϕ

Algorithm 1. Generate contour merging ground truth

Results

Implementation details

In our experiments, we optimized the hyperparameters to achieve robust segmentation performance. Mask R-CNN[24] with ResNet-50[48] and Swin Transformer[49] backbones was trained using an initial learning rate of 0.001, which was decayed by a factor of 0.1 every 10 epochs. The Adam optimizer was employed with parameters β_1 = 0.9 and β_2 = 0.999 to ensure stable convergence. A batch size of 8 was used to balance computational efficiency and model generalization, while a weight decay of 0.0001 helped prevent overfitting. The training process spanned 100 epochs, incorporating various data augmentation techniques such as horizontal flips, vertical flips, and random rotations within a $\pm 10^\circ$ range to improve model robustness. For instance segmentation, the YOLOv11 backbone[50] was employed with an initial learning rate of 0.0025, incorporating a linear warm-up strategy during the first three epochs to facilitate stable optimization. The SGD optimizer was used with a momentum of 0.9, a batch size of 8, and a weight decay of 0.0005. The model was trained for 100 epochs with data augmentation techniques, including color jittering to enhance color diversity, random cropping to improve spatial robustness, and mosaic augmentation to introduce more complex training samples. Detectree2[34] was trained for 100 epochs using a learning rate of 0.0035 to optimize tree segmentation performance.

The proposed method followed a two-phase training strategy. In the first phase, the edge detection network was trained for 200 epochs using the Adam optimizer with a learning rate of 0.01 and a weight decay of 0.0005. This phase generated the initial edge maps required for subsequent processing. In the second phase, the GCN network responsible for contour merging was trained for 200 epochs with a learning rate of 0.1 and a weight decay of 0.005, enabling accurate contour refinement for improved segmentation quality. All experiments were conducted on a workstation equipped with an NVIDIA RTX 4080 Ti GPU, ensuring efficient training and inference.

Evaluation metrics

As the proposed method has a classification step to decide whether the two adjacent nodes in the graph should be merged, therefore, in our experiments, in addition to the metrics used for instance segmentation evaluation, we also employ the accuracy metric for assessing the classification step.

The accuracy metric is defined as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{18}$$

where: TP (true positives): correctly classified positive samples. TN (true negatives): correctly classified negative samples. FP (false positives): negative samples incorrectly classified as positive. FN (false negatives): positive samples incorrectly classified as negative.

This metric reflects the effectiveness of the model in distinguishing object boundaries and ensuring proper contour association. High classification accuracy indicates that the model correctly predicts relationships between pair of contours, leading to more precise segmentation.

Concerning the instance segmentation evaluation, the model's performance is assessed using the Average Precision (AP) metric. The primary evaluation metric is Average Precision (AP), which is based on the precision-recall curve and the Intersection over Union (IoU) threshold.

Precision and recall are defined as:

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN} \tag{19}$$

The average precision (AP) is computed as the area under the precision-recall curve:

$$AP = \int_0^1 P(R)\, dR \tag{20}$$

In discrete form, it is approximated as:

$$AP = \sum_{i=1}^{N} (R_i - R_{i-1}) P_i \tag{21}$$

where N is the number of recall points. In our study, the AP at different IoU thresholds including AP@50 (IoU ≥ 0.5) and AP@70 (IoU ≥ 0.70), along with the average Average Precision (AP) across multiple thresholds will be computed. Higher AP score signifies better segmentation performance, demonstrating the model's robustness in distinguishing instances with varying shapes, sizes, and occlusions. Together, these metrics help to understand the strengths and weaknesses of the model.

Ablation study

Evaluation of the impact of patch size

As mentioned earlier, we extracted patches within object contours to measure similarity between graph nodes. To evaluate the effect of patch size on model performance, we conducted an extensive analysis across various

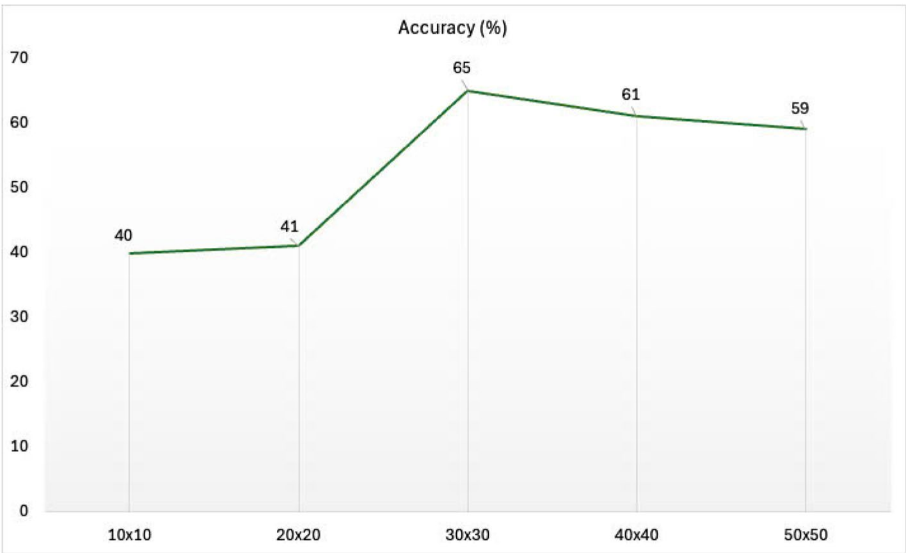

Fig. 8. Classification results with different patch sizes in ForestSeg-T1 dataset.

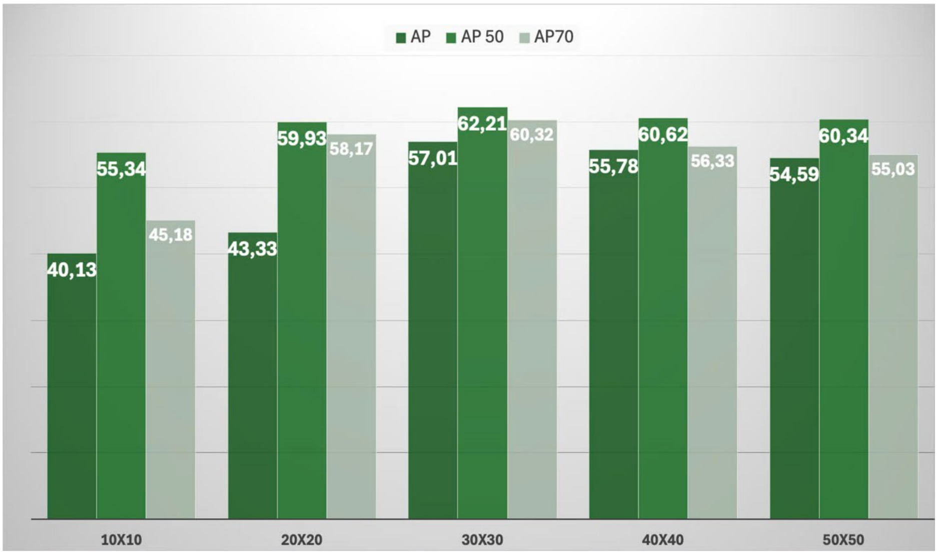

Fig. 9. Instance segmentation results with different patch sizes in ForestSeg-T1 dataset.

Contour generation method	Classification accuracy (%)	AP	AP_{50}	AP_{70}
	(ForestSeg-T1 contour pairs)			
PiDiNet[35]	60.22	50.77	55.21	53.97
DexiNed[51]	62.58	54.14	58.66	54.09
EDTER[7]	**65.37**	**57.01**	**62.21**	**55.32**

Table 3. Segmentation performance with different contour generation methods on ForestSeg-T1 (Best results for each metric are shown in bold).

patch dimensions. Figure 8 illustrates the classification accuracy, while Fig.9 displays the instance segmentation results under different patch sizes. Among patch sizes ranging from 10×10 to 50×50 pixels, a patch size of 30×30 yields the highest classification accuracy (65%), highlighting its effectiveness in distinguishing instances. Furthermore, this patch size also achieves the best instance segmentation performance, with an AP of 57.01%, AP@50 of 62.21%, and AP@70 of 60.32%, outperforming all other configurations.

Although larger patch sizes such as 40×40 and 50×50 also achieve relatively high scores, their performance consistently falls short of the 30×30 configuration across all metrics. For example, the 40×40 patch yields 61% accuracy and an AP of 55.78%, while the 50×50 patch achieves 59% accuracy and an AP of 54.59%. These results suggest that overly large patches may include irrelevant background information, diminishing both classification and segmentation performance. Conversely, smaller patches such as 10×10 and 20×20 show significantly lower effectiveness. The 10×10 patch size obtains only 40% accuracy, with an AP of 40.13%, AP@50 of 55.34%, and AP@70 of 45.18%, indicating insufficient contextual information for robust similarity estimation. The 20×20 patch performs slightly better, with 41% accuracy and an AP of 43.33%, but still lags behind larger patches. Based on these findings, we conclude that the 30×30 patch size offers the best trade-off between capturing meaningful features and minimizing noise. As such, we adopt $p = 30$ as the standard patch size in all subsequent experiments to ensure optimal classification and segmentation performance.

Evaluation of different contour generation methods

To evaluate the contribution of the contour generation step in our pipeline, we conducted a comparative study using three deep edge detection approaches: PiDiNet[35], DexiNed[51], EDTER[7]. These methods were employed exclusively in the contour generation stage, where their role is to divide tree canopies into finer candidate instances. Following this step, all subsequent components of the proposed framework were kept identical, including the GCN-based contour merging module, which aggregates contours with similar structural and appearance characteristics to form final tree instances. Table 3 presents the quantitative performance of the three contour generation strategies in terms of Classification Accuracy and Average Precision metrics.

As shown in Table 3, the proposed EDTER-based contour generation consistently outperforms PiDiNet and DexiNed across all evaluation metrics. In particular, EDTER achieves the highest Classification Accuracy (65.37%) and AP (57.01%), indicating its superior capability in producing more discriminative and structurally consistent contours. Compared to PiDiNet[35], EDTER yields an improvement of 5.15 percentage points in classification accuracy and 6.24 points in AP, demonstrating that more precise contour initialization significantly

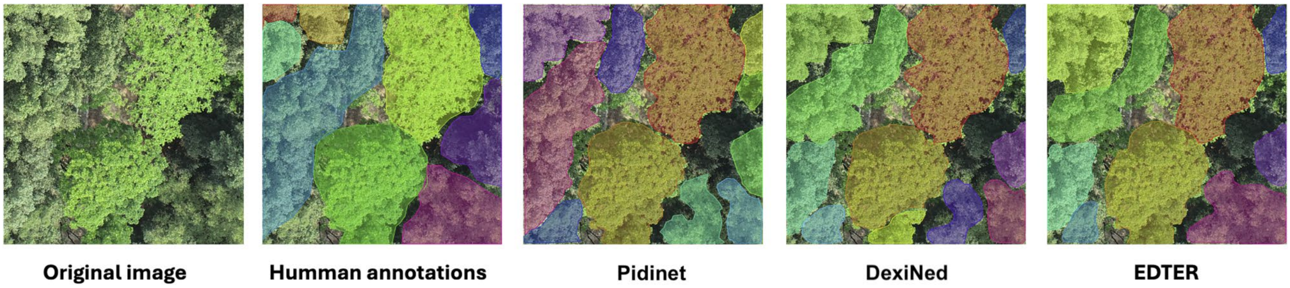

Fig. 10. Visualizations of segmentation results with three contour generation methods: Pidinet[35], DexiNed[51], EDTER[7] in ForestSeg-T1 dataset.

Features	Classification accuracy (%)	AP	AP_{50}	AP_{70}
	(ForestSeg-T1 contour pairs)			
W/o merging	35.36	20.03	35.38	17.52
W/o similarity on appearance feature	43.01	44.35	46.28	40.03
W/o area	**65.37**	**57.01**	**62.21**	**55.32**
W/o extent	63.14	50.76	59.31	49.33
W/o solidity	64.02	51.22	58.39	50.03
W/o aspect ratio	62.17	53.41	55.43	50.23
W/o deviation	52.31	50.50	52.08	40.02
All	57.94	52.50	59.37	50.49

Table 4. Segmentation results with different feature in ForestSeg-T1 (best results are in bold).

enhances the downstream instance construction process. DexiNed performs better than PiDiNet but still lags behind EDTER, especially under stricter matching criteria such as AP_{70}.

These results confirm that the quality of the contour generation stage plays a critical role in the overall performance of the proposed framework, and that the EDTER-based strategy provides more reliable boundary representations for graph-based instance merging. Figure 10 illustrates some segmentation results with different contour generation methods.

Evaluation of the role of features

In the proposed method, a graph is constructed in which each node represents a contour, and each edge encodes the relationship between a pair of contours. For each node, a five-dimensional shape feature vector including area, extent, solidity, aspect ratio and deviation is computed, while appearance features are used to determine the weights of the graph edges. Starting from contours initialized by the contour generation method, contours are subsequently merged to produce more accurate instance segmentation results. To evaluate the effectiveness of the merging step and the contribution of each feature, we conduct an ablation study to analyze the impact of individual components in the proposed model. This analysis systematically removes or modifies specific components to evaluate their impact on overall performance. By isolating individual features, we aim to determine their relative importance and understand how each element contributes to the accuracy of the instance segmentation task.

Table 4 presents classification accuracy and instance segmentation results using different features. Initially, we evaluate the performance of the merging process. As discussed, the effectiveness of our method relies on two main components: contour extraction and contour merging. The contour extraction module is responsible for distinguishing different canopies, but it often fragments a single canopy into multiple smaller parts. Therefore, relying solely on this module leads to reduced performance. As shown in Table 4, for the segmentation task, omitting the merging process results in the most significant performance drop with AP, AP@50, and AP@70 decreasing to just 20.03%, 35.38%, and 17.52%, respectively. This highlights the critical role of contour merging in the overall performance.

We first assessed the impact of the similarity score, which determines the relationship between nodes (i.e., contours) in the graph. Without this similarity score, the graph lacks edge weights, making the merging process less informed. Nonetheless, merging without similarity still better than without merging with 43.01% classification accuracy, 44.35%, 46.28%, and 40.03% of AP, AP@50, and AP@70 respectively. However, these values remain lower than the full model that incorporates all features. This demonstrates that the similarity score is crucial for improving recognition and segmentation performance by helping prevent over-segmentation (i.e. splitting a tree into multiple parts) and under-segmentation (i.e. merging neighboring trees incorrectly).

Then, we evaluated the importance of shape features extracted for node representation including area, extent, solidity, aspect ratio, and deviation features. Experimental results show that using four features (extent, solidity,

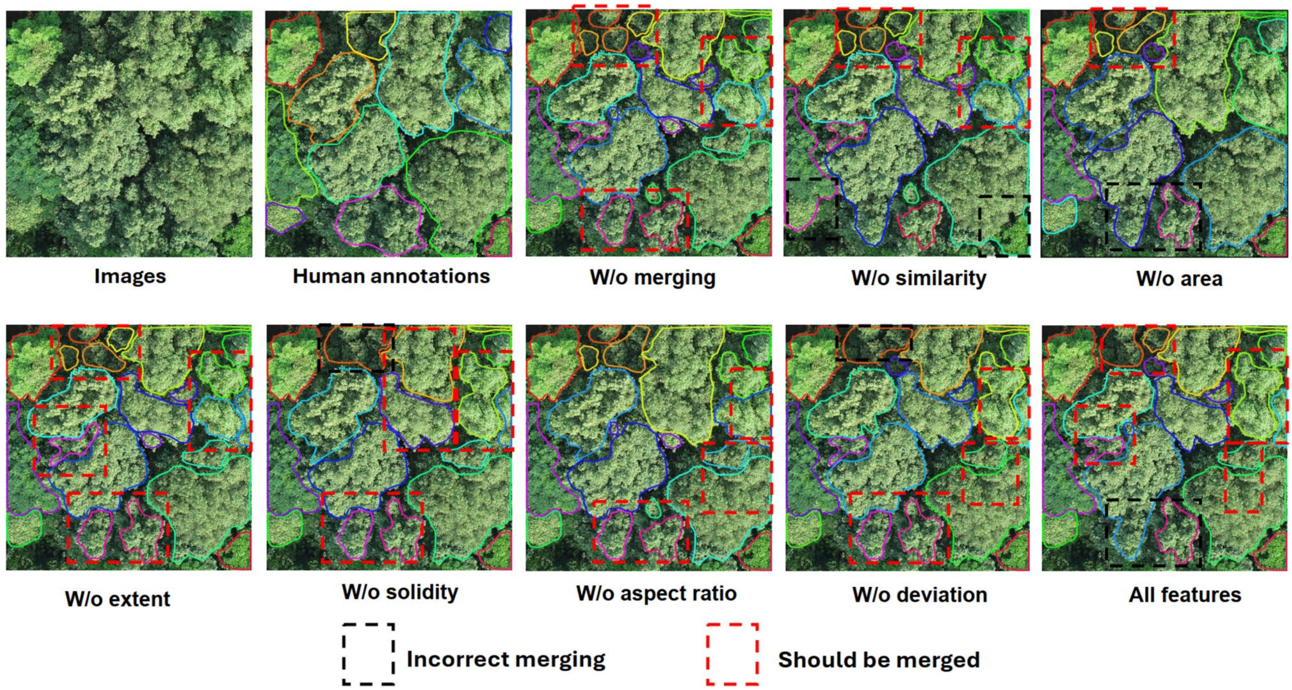

Fig. 11. Visualization of instance segmentation results with different features in ForestSeg-T1 dataset (each color illustrates for one instance).

Method	Inference Time (ms)	AP (%)	AP$_{50}$ (%)	AP$_{70}$ (%)
Mask R-CNN (ResNet-50)[48]	8.3	30.63	46.23	26.17
Mask R-CNN (Swin-T)[49]	10.8	56.72	60.12	54.64
Mask2Former[52]	9.6	20.12	25.67	10.55
Detectree2[34]	11.4	22.33	49.71	20.22
YOLOV11[50]	7.5	38.30	52.78	33.51
Ours	**6.2**	**57.01**	**62.21**	**55.32**

Table 5. Comparison of segmentation performance between the proposed method and the state-of-the-art approaches in ForestSeg-T1. Best results for each metric are shown in bold.

aspect ratio, and deviation) results the highest performance, with a classification accuracy of 65.37%, AP of 57.01%, AP@50 of 62.21%, and AP@70 of 55.32%. This suggests that the area feature might introduce noise or offer limited useful information in UAV-based tree segmentation, likely due to significant variations in canopy size depending on flight altitude and viewing angle. Moreover, the study area is a natural forest containing diverse tree species at different growth stages. Therefore, tree crown sizes vary substantially.

When the extent feature is removed, the model attains a classification accuracy of 63.14%, however, the AP, AP@50, and AP@70 values are slightly lower than those achieved when using all features, indicating that while extent is not critical, it may offer moderate benefits for precise instance localization. Similarly, excluding solidity results in a classification accuracy of 64.02%, outperforming with using all features in terms of classification. However, the segmentation scores remain close to or slightly below the full features score values, suggesting that solidity plays a limited role in delineating canopy boundaries but may still support overall recognition. In the case of aspect ratio, the model achieves 62.17% accuracy and comparable AP scores, implying that this feature has a marginal influence on the model's decision-making process. While it may contribute to shape characterization, its absence does not cause a significant degradation in performance. By contrast, removing the deviation feature leads to the most notable decline in all evaluation metrics compared to the full feature set. The model achieves only 52.31% classification accuracy, with AP dropping to 50.50%, AP@50 to 52.08%, and AP@70 to 40.02%. This degradation is most apparent in AP@70, suggesting that deviation play an important role for achieving higher segmentation precision. Figure 11 illustrates the intermediate results of the proposed method.

Overall, the results demonstrate that all components contribute to the performance of the proposed method. Regarding the shape features, the ablation study highlights the significant impact of the four shape features–extent, solidity, aspect ratio, and deviation–on the model's effectiveness. As a result, these features are retained for all subsequent experiments.

Comparison with the-state-of-the-art methods

We compare the proposed method against the state-of-the-art segmentation models, including Mask-RCNN (ResNet-50)[48], Mask-RCNN (SwinT)[49], Detectree2[34], YOLOV11[50] and Mask2Former[52]. YOLOv11 is a recent one-stage framework that supports instance segmentation by jointly predicting bounding boxes and pixel-level masks. These models are trained and evaluated on ForestSeg-T1 set of ForestSeg. Results are shown in Table 5. The results highlight the superior efficiency and accuracy of the proposed method. It achieves the highest segmentation performance across all three accuracy metrics, with an AP of 57.01%, AP@50 of 62.21%, and AP@70 of 55.32%, surpassing all competing models.

Among the models, Mask-RCNN (SwinT)[49] achieves a good performance with an AP of 56.72%, AP@50 of 60.12%, and AP@70 of 54.64%. While these results indicate competitive performance, a main drawback of Mask-RCNN (SwinT) lies in its substantially higher computational complexity. Specifically, it exhibits an inference time of 10.8 ms, making it significantly less efficient for real-time applications, particularly in scenarios with constrained computing resources such as embedded systems, UAVs, or mobile devices. YOLOV11[50], a model specifically designed for high-speed inference, processes images at 7.5 ms per frame. While this is relatively fast, it still lags behind the proposed method in both efficiency and accuracy with an AP of 38.30%, AP@50 of 52.78%, and AP@70 of 33.51%, indicating a trade-off where speed comes at the expense of segmentation quality. Similarly, Detectree2[34], despite being a highly specialized model for tree segmentation, requires an extensive of 11.4 ms inference time and produces low performance. Among these methods, Mask2Former achieves the lowest performance, with an AP of 20.12%, AP@50 of 25.67%, and AP@70 of 10.55%. These results indicate that Mask2Former may struggle to handle fine and ambiguous boundaries, which are prevalent in tree instance segmentation tasks.

Contrary to these methods, the proposed method demonstrates remarkable computational efficiency.This drastic reduction in model size translates into faster inference speeds, with the proposed approach achieving an impressive 6.2 ms per image, making it the most efficient among all compared models.

These results emphasize the capability of the proposed method to achieve the state-of-the-art accuracy while maintaining a lightweight model structure and superior inference speed. This balance between accuracy and efficiency is particularly crucial for real-time applications where computational resources are limited. By significantly reducing both the model size and inference time without compromising segmentation quality, the proposed approach presents a promising alternative to existing segmentation models.

Figure 12 illustrates the instance segmentation results, the visualized image was selected in ForestSeg-T1. The visual analysis highlights the superior object delineation, sharper boundary detection, and reduced false positives achieved by the proposed method.

Evaluation of segmentation performance in different intervals

Understanding the influence of temporal variations on instance segmentation performance is critical for robust forest monitoring applications. Environmental conditions–such as lighting, shadow patterns, and seasonal changes—can significantly affect the appearance of UAV-acquired RGB imagery. In this section, we evaluate our segmentation model's performance across different times of day and seasonal periods to assess its robustness under varying illumination and phenological conditions. By comparing metrics such as Average Precision (AP), AP_{50}, and AP_{70} on datasets collected at different times, we aim to quantify how changes in factors like sun angle, weather conditions, and seasonal foliage variations impact the segmentation accuracy. This evaluation provides insight into the model's generalization and its practical applicability for long-term forest inventory and health monitoring.

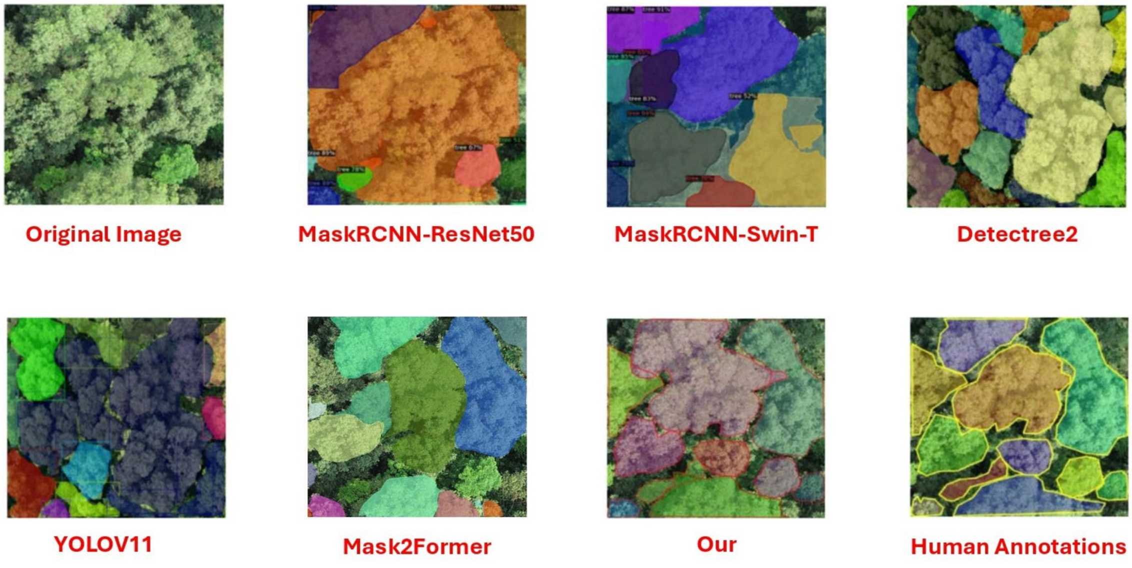

Fig. 12. Instance segmentation results of the SOTA methods in ForestSeg-T1 dataset.

Forest Name	Classification accuaracy (%)	AP	AP_{50}	AP_{70}
ForestSeg-T1	65.37	57.01	62.21	55.32
ForestSeg-T2	69.23	59.72	62.11	55.12
ForestSeg-T3	59.01	54.92	57.28	50.01
ForestSeg-T4	71.32	60.82	64.71	57.13

Table 6. Experiment results obtained by the proposed method in different sets of ForestSeg captured in different intervals.

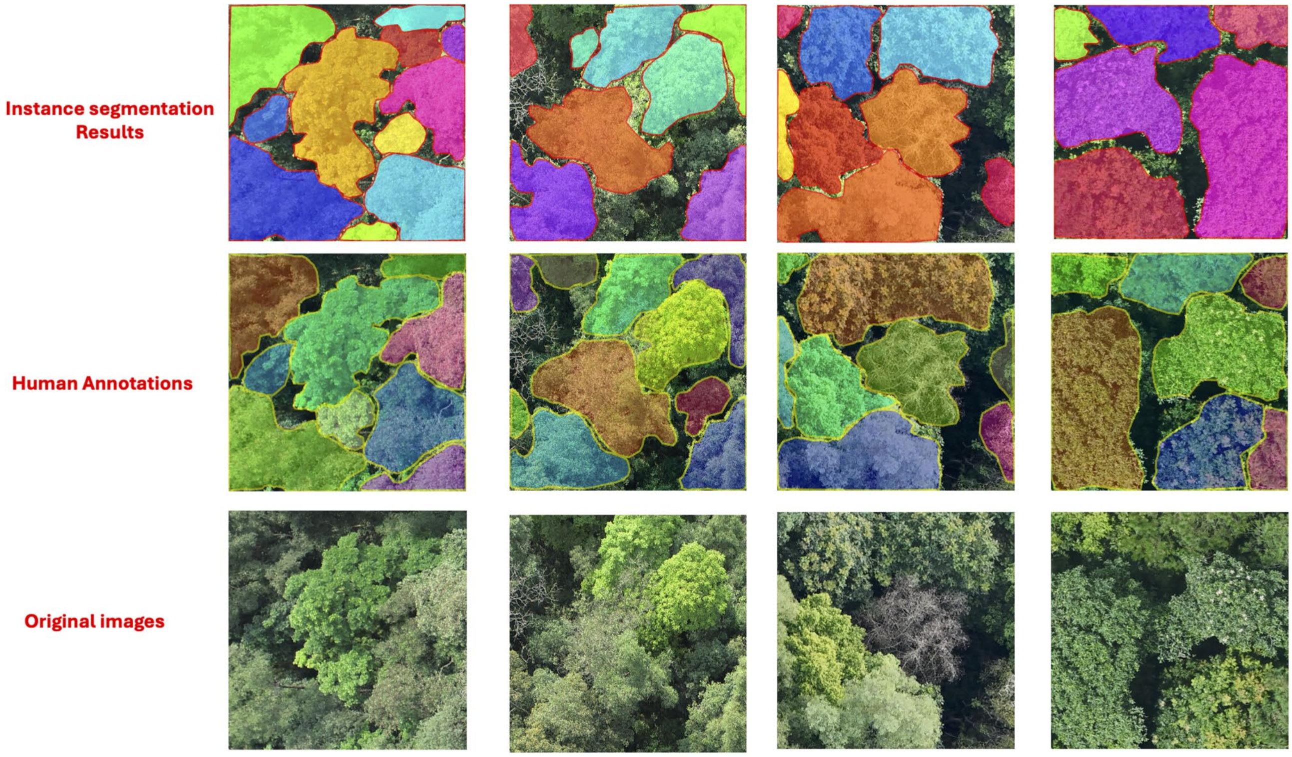

Fig. 13. Visualization of instance segmentation results in four different intervals in ForestSeg dataset (ForestSeg-T1, ForestSeg-T2, ForestSeg-T3, ForestSeg-T4 respectively from left to right).

Table 6 presents the experimental results for different forest segmentation intervals, denoted as ForestSeg-T1 to ForestSeg-T4. Among the four tested intervals, ForestSeg-T4 achieves the highest classification accuracy at 71.32%, followed by ForestSeg-T2 (69.23%), ForestSeg-T1 (65.37%), and ForestSeg-T3 (59.01%). The superior classification performance of ForestSeg-T4 suggests that this interval provides the most informative features for distinguishing forest segments.

In terms of segmentation performance, the proposed method obtains the best results on ForestSeg-T4, with an AP of 60.82%, AP@50 of 64.71%, and AP@70 of 57.13%.

The results for ForestSeg-T2 follows with an AP of 59.72%, AP@50 of 62.11%, and AP@70 of 55.12%, while those achieved for ForestSeg-T1 are AP of 57.01%, AP@50 of 62.21%, and AP@70 of 55.32%. Notably, results on ForestSeg-T3 demonstrates the lowest performance in both classification and segmentation, with an AP of 54.92%, AP@50 of 57.28%, and AP@70 of 50.01%. The reason is that in ForestSeg-T4, images were captured using the DJI Air 3 drone with the 3× telephoto camera, allowing for finer spatial resolution of individual tree crowns even from a flight altitude of 100 meters. This contrasts with ForestSeg-T2 and T3, which were acquired using the standard 1× camera. Although results for ForestSeg-T2 shows relatively good classification accuracy, the AP scores for this set remain lower than those for ForestSeg-T4, possibly due to less distinguishable canopy structures. In ForestSeg-T3 set, the proposed method yielded the lowest overall performance. The relatively lower performance observed on ForestSeg-T3 can be explained by its distinct acquisition conditions compared to the other subsets. As shown in Table 1, ForestSeg-T3 was captured at a different time of day, between 12:30 and 13:30, which corresponds to strong midday illumination. In addition, it was acquired at a higher flight altitude of 150 m and covers a larger area, leading to increased scale variation and reduced boundary contrast between adjacent tree crowns.

Figure 13 illustrates instance segmentation outputs across four different intervals (ForestSeg-T1 to ForestSeg-T4), covering a range of spatial and textural variations. Each subfigure compares the segmentation

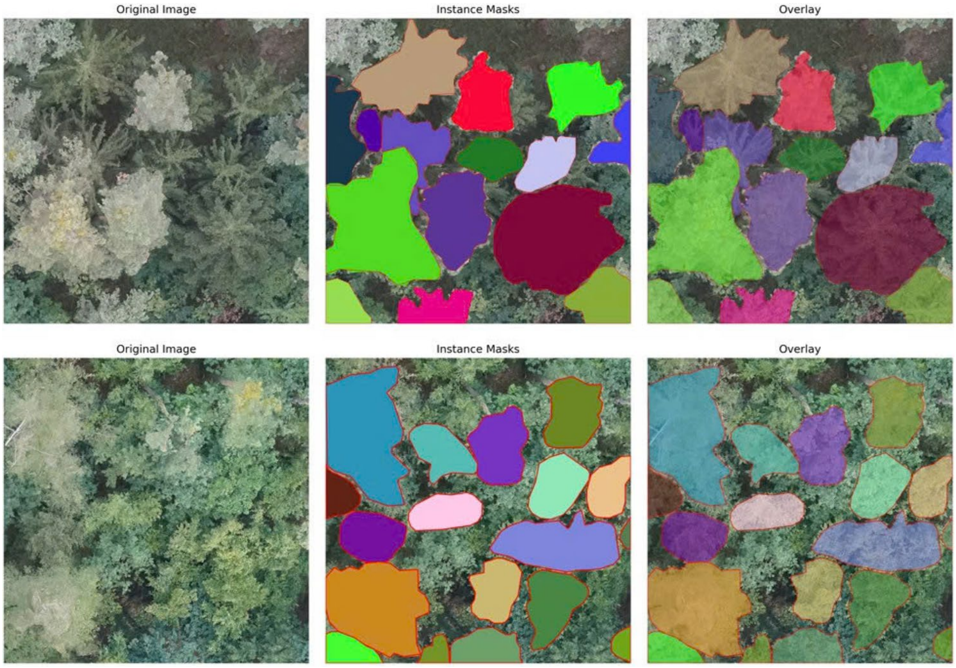

Fig. 14. Visualization of instance masks and their annotations in BAMFORESTS dataset.

Method	AP (%)	AP$_{50}$ (%)	AP$_{70}$ (%)
Mask R-CNN (ResNet-50)[48]	40.01	70.75	41.35
Mask R-CNN (Swin-T)[49]	42.17	72.14	43.32
Detectree2[34]	38.73	64.22	40.01
YOLOV11[50]	41.21	71.02	45.21
Ours	**53.21**	**73.14**	**43.24**

Table 7. Comparison of segmentation performance between the proposed method and the state-of-the-art approaches in BAMFORESTS dataset. Best results for each metric are shown in bold.

results produced by our method with the corresponding human annotations and original images. By showcasing multiple instances from different environments, this visualization highlights the robustness, consistency, and accuracy of our approach in handling complex and diverse forest scenes. These qualitative results complement our quantitative evaluation and provide further evidence of the method's practical effectiveness.

Evaluation on a benchmark dataset

To comprehensively evaluate the effectiveness of our proposed method, we conduct experiments by comparing it against state-of-the-art approaches on the BAMFORESTS dataset[37], which was originally designed for forest canopy segmentation tasks. The detailed characteristics of this dataset are summarized in Table 2. Since BAMFORESTS and ForestSeg share similar properties in terms of canopy structure, spatial resolution, and annotation format, this dataset serves as an appropriate benchmark for testing the transferability of our approach. There are two versions of this dataset: one with image dimensions of 1024×1024 pixels and another with 2048×2048 pixels. For all experiments conducted in this study, we utilized the 1024×1024 version for fair evaluation with ForsetSeg dataset. Figure 14 illustrates some samples with their annotations in this dataset.

We evaluate the effectiveness of our proposed method on the BAMFORESTS dataset to comprehensively compare it with several state-of-the-art segmentation approaches. As reported in Table 7, our method achieves the best overall performance, obtaining 53.21%, 73.14%, and 43.24% in AP, AP@50, and AP@70, respectively. It is worth noting that on the BAMFORESTS dataset, TreeCoG achieves the best AP and AP@50, while its AP@70 of 43.24% is slightly lower than the AP@70 of 45.21% achieved by YOLOv11. This difference may be related to the characteristics of the BAMFORESTS dataset, which includes many trees with relatively regular crown shapes, clearer boundaries, and lower inter-instance overlap. Under such conditions, bounding-box-oriented detectors such as YOLOv11 may be more effective at achieving very tight localization.

These results clearly indicate that our model not only delivers high segmentation accuracy but also demonstrates strong robustness and generalization capability when applied to forest scenes containing trees with diverse shapes, densities, and ecological characteristics. The consistent improvements across all evaluation

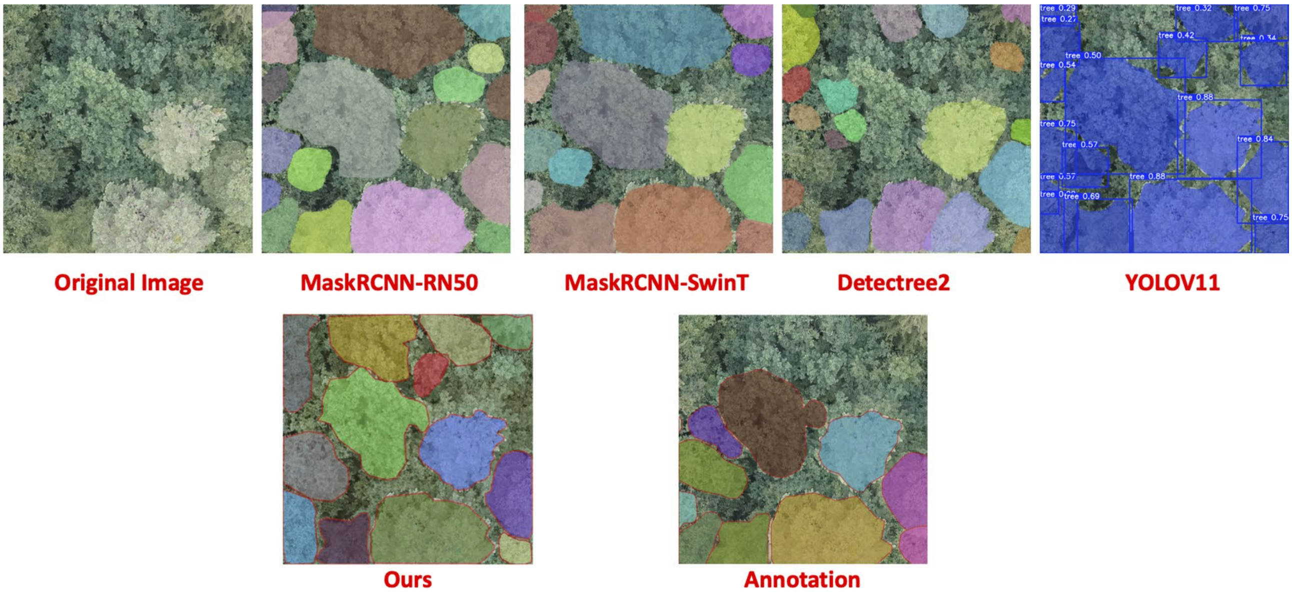

Fig. 15. Visualization of instance segmentation results with SOTA methods in BAMFORESTS dataset.

Train/test	AP	AP@50	AP@70
ForestSeg-T1 / ForestSeg-T1	57.01	62.21	55.32
BAMFORESTS / BAMFORESTS	53.21	73.14	43.24
ForestSeg-T1 / BAMFORESTS	24.14	31.35	17.86

Table 8. Segmentation performance in cross-dataset evaluation.

metrics confirm that our approach effectively captures both local geometric structures and global contextual information within complex forest environments.

Representative visualizations of the segmentation results from different competing methods on the BAMFORESTS dataset are presented in Figure 15, highlighting the superior ability of our model to delineate fine-grained tree boundaries and preserve structural details under challenging conditions.

Cross-dataset evaluation

To further assess the generalization capability of the proposed method, we conduct additional cross-dataset experiments between ForestSeg-T1 and BAMFORESTS. These experiments aim to better understand the impact of domain shift in UAV forest instance segmentation. The results in Table 8 highlight the significant impact of domain shift on instance segmentation performance in UAV forest imagery. When training and testing are conducted on the same dataset, the model achieves strong performance on both ForestSeg-T1 and BAMFORESTS, with AP values of 57.01% and 53.21%, respectively. This indicates that the model is effective at learning dataset-specific characteristics under in-domain conditions.

However, when the model trained on ForestSeg-T1 is evaluated on BAMFORESTS, performance drops substantially, with AP decreasing to 24.14%, AP@50 to 31.35%, and AP@70 to 17.86%. This degradation reflects the considerable domain gap between the two datasets, which arises from differences in forest structure, tree species composition, crown size distribution, imaging conditions, and annotation styles.

Therefore, in future work, we plan to investigate domain adaptation methods[53,54] to enhance the robustness of tree instance segmentation across different forest domains.

Conclusion and future works

In this paper, we introduced a novel method for tree instance segmentation to support forest monitoring. Our approach leverages a Deep Edge Detection Network to effectively distinguish individual tree canopies and employs a Graph Convolutional Network (GCN) to merge contours belonging to the same tree. Additionally, we proposed a new dataset with various intervals to evaluate our method comprehensively. Experimental results demonstrate the superior performance of TreeCoG in tree instance segmentation, achieving 57.01% AP, 62.21% AP@50, and 55.32% AP@70 on the ForestSeg dataset, and 53.21% AP, 73.14% AP@50, and 43.23% AP@70 on the BAMFORESTS dataset, respectively.

Our method outperforms the state-of-the-art approaches in terms of instance segmentation accuracy, inference time, and model efficiency, highlighting its suitability for large-scale forest monitoring applications. For future work, we aim to further refine the model by incorporating self-supervised learning techniques to enhance feature and explore adaptive patch selection strategies to improve contour merging accuracy. Furthermore, based on the segmentation results, plant identification will be performed to build a forest biodiversity map. Extending the dataset with more diverse forest environments and testing the method on high-resolution aerial imagery will also be key directions for future research.

Data availability

ForestSeg dataset used in this paper is available from the corresponding author upon reasonable request while BAMFORESTS dataset is available at [https://www.dlr.de/en/eoc/about-us/remote-sensing-technology-institute/photogrammetry-and-image-analysis/public-datasets/bamforests].

Received: 6 November 2025; Accepted: 13 January 2026
Published online: 20 January 2026

References

1. Baltrėnaitė, E., Baltrėnas, P. & Lietuvninkas, A. *The Sustainable Role of the Tree in Environmental Protection Technologies* (Springer, 2016).
2. Cheng, J., Deng, C., Su, Y., An, Z. & Wang, Q. Methods and datasets on semantic segmentation for unmanned aerial vehicle remote sensing images: A review. *ISPRS J. Photogramm. Remote Sens.* **211**, 1–34 (2024).
3. Näsi, R. et al. Using UAV-based photogrammetry and hyperspectral imaging for mapping bark beetle damage at tree-level. *Remote Sens.* **7**, 15467–15493 (2015).
4. Sun, P., Yuan, X. & Li, D. Classification of individual tree species using UAV lidar based on transformer. *Forests* **14**, 484 (2023).
5. Marcello, J. et al. Performance of individual tree segmentation algorithms in forest ecosystems using UAV lidar data. *Drones* https://doi.org/10.3390/drones8120772 (2024).
6. Noss, R. F. Assessing and monitoring forest biodiversity: A suggested framework and indicators. *For. Ecol. Manag.* **115**, 135–146. https://doi.org/10.1016/S0378-1127(98)00394-6 (1999).
7. Pu, M., Huang, Y., Liu, Y., Guan, Q. & Ling, H. Edter: Edge detection with transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1402–1412 (2022).
8. Firoze, A., Wingren, C., Yeh, R. A., Benes, B. & Aliaga, D. Tree instance segmentation with temporal contour graph. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2193–2202. https://doi.org/10.1109/CVPR52729.2023.00218 (2023).
9. Xue, Y. et al. Fmtrack: Frequency-aware interaction and multi-expert fusion for rgb-t tracking. *IEEE Trans. Circuits Syst. Video Technol.* **35**, 1–1. https://doi.org/10.1109/TCSVT.2025.3601598 (2025).
10. Xue, Y. et al. Target-distractor aware UAV tracking via global agent. *IEEE Trans. Intell. Transport. Syst.* **26**, 16116–16127. https://doi.org/10.1109/TITS.2025.3581391 (2025).
11. Zheng, J., Yuan, S., Li, W., Fu, H. & Yu, L. A review of individual tree crown detection and delineation from optical remote sensing images: Current progress and future. *IEEE Geosci. Remote Sens. Mag.* **13**, 209–236. https://doi.org/10.1109/MGRS.2024.3479871 (2025).
12. Zheng, J. et al. Growing status observation for oil palm trees using unmanned aerial vehicle (UAV) images. *ISPRS J. Photogramm. Remote Sens.* **173**, 95–121. https://doi.org/10.1016/j.isprsjprs.2021.01.008 (2021).
13. Zheng, J. et al. Cross-regional oil palm tree counting and detection via a multi-level attention domain adaptation network. *ISPRS J. Photogramm. Remote Sens.* **167**, 154–177. https://doi.org/10.1016/j.isprsjprs.2020.07.002 (2020).
14. Zheng, J. et al. Multisource-domain generalization-based oil palm tree detection using very-high-resolution (vhr) satellite images. *IEEE Geosci. Remote Sens. Lett.* **19**, 1–5. https://doi.org/10.1109/LGRS.2021.3117866 (2022).
15. Liu, J.-J., Hou, Q., Cheng, M.-M., Feng, J. & Jiang, J. Poolnet+: Exploring the potential of pooling for salient object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**, 487–504. https://doi.org/10.1109/TPAMI.2022.3156828 (2022).
16. Li, Y., Wang, H. & Katsaggelos, A. Cpdr: Towards highly-efficient salient object detection via crossed post-decoder refinement. In *35th British Machine Vision Conference (BMVC)* (BMVA, 2024).
17. Lv, X. et al. Deep merge: Deep-learning-based region merging for remote sensing image segmentation. *IEEE Trans. Geosci. Remote Sens.* **63**, 1–14. https://doi.org/10.1109/TGRS.2025.3544549 (2025) (art no. 5614120).
18. Sozio, A. et al. An innovative SAM-VIT based tool for the automatic detection of litter items on sandy beaches. In *EGU General Assembly 2024*, EGU24–8657. https://doi.org/10.5194/egusphere-egu24-8657 (2024).
19. Teng, M. et al. Assessing SAM for tree crown instance segmentation from drone imagery. *arXiv preprint* arXiv:2503.20199 (2025).
20. Chen, K. et al. Rsprompter: Learning to prompt for remote sensing instance segmentation based on visual foundation model. *IEEE Trans. Geosci. Remote Sens.* **62**, 1–17. https://doi.org/10.1109/TGRS.2024.3356074 (2024).
21. Ecke, S. et al. Towards operational UAV-based forest health monitoring: Species identification and crown condition assessment by means of deep learning. *Comput. Electron. Agric.* **219**, 108785. https://doi.org/10.1016/j.compag.2024.108785 (2024).
22. Salem, M., Gomaa, A. & Tsurusaki, N. Detection of earthquake-induced building damages using remote sensing data and deep learning: A case study of Mashiki Town, Japan. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. 5257–5260. https://doi.org/10.1109/IGARSS52108.2023.10282439 (IEEE, 2023).
23. Cui, X. & Zhang, L. CADSM: Crossing-attention dual-scanning Mamba for multimodal remote sensing image classification. *J. Appl. Remote Sens.* **20**, 021406. https://doi.org/10.1117/1.JRS.20.021406 (2025).
24. He, K., Gkioxari, G., Dollár, P. & Girshick, R. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*. 2961–2969 (2017).
25. Safonova, A., Guirado, E., Maglinets, Y., Alcaraz-Segura, D. & Tabik, S. Olive tree biovolume from UAV multi-resolution image segmentation with mask r-cnn. *Sensors* https://doi.org/10.3390/s21051617 (2021).
26. Lv, L. et al. A deep learning network for individual tree segmentation in UAV images with a coupled cspnet and attention mechanism. *Remote Sens.* https://doi.org/10.3390/rs15184420 (2023).
27. Han, H., Zhu, F., Zhu, B. & Wu, H. Target detection of remote sensing image based on an improved yolov5. *IEEE Geosci. Remote Sens. Lett.* **20**, 1–5 (2023).
28. Straker, A. et al. Instance segmentation of individual tree crowns with yolov5: A comparison of approaches using the for instance benchmark lidar dataset. *ISPRS Open J. Photogramm. Remote Sens.* **9**, 100045. https://doi.org/10.1016/j.ophoto.2023.100045 (2023).
29. Ziyi Sun, M. Z., Xue, B. & Schindler, J. Yolov8e: An efficient yolov8 method for instance segmentation of individual tree crowns in Wellington City, New Zealand. *J. R. Soc. New Zealand* 1–26. https://doi.org/10.1080/03036758.2024.2424802 (2024).
30. He, A. et al. Alss-yolo: An adaptive lightweight channel split and shuffling network for TIR wildlife detection in UAV imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* (2024).
31. Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III* . Vol. 18. 234–241 (Springer, 2015).

32. Xie, Y. et al. Instance segmentation and stand-scale forest mapping based on UAV images derived rgb and chm. *Comput. Electron. Agric.* **220**, 108878 (2024).
33. Zhang, X., Shen, J., Hu, H. & Yang, H. A new instance segmentation model for high-resolution remote sensing images based on edge processing. *Mathematics* https://doi.org/10.3390/math12182905 (2024).
34. Ball, J. G. et al. Accurate delineation of individual tree crowns in tropical forests from aerial rgb imagery using mask r-cnn. *Remote Sens. Ecol. Conserv.* **9**, 641–655 (2023).
35. Su, Z. et al. Pixel difference networks for efficient edge detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5117–5127 (2021).
36. Mowla, M. N., Asadi, D., Tekeoglu, K. N., Masum, S. & Rabie, K. Uavs-ffdb: A high-resolution dataset for advancing forest fire detection and monitoring using unmanned aerial vehicles (uavs). *Data Brief* **55**, 110706. https://doi.org/10.1016/j.dib.2024.110706 (2024).
37. Troles, J., Schmid, U., Fan, W. & Tian, J. Bamforests: Bamberg benchmark forest dataset of individual tree crowns in very-high-resolution UAV images. *Remote Sens.* https://doi.org/10.3390/rs16111935 (2024).
38. Puliti, S. et al. For-instance: A UAV laser scanning benchmark dataset for semantic and instance segmentation of individual trees. *arXiv preprint* arXiv:2309.01279 (2023).
39. Cheng, D. et al. Treescope: An agricultural robotics dataset for lidar-based mapping of trees in forests and orchards. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*. 14860–14866 (IEEE, 2024).
40. Reiersen, G. et al. Reforestree: A dataset for estimating tropical forest carbon stock with deep learning and aerial imagery. *Proc. AAAI Conf. Artif. Intell.* **36**, 12119–12125 (2022).
41. Troles, J., Nieding, R., Simons, S. & Schmid, U. Task planning support for arborists and foresters: Comparing deep learning approaches for tree inventory and tree vitality assessment based on UAV-data. In *International Conference on Innovations for Community Services*. 103–122 (Springer, 2023).
42. Sharma, R., Saqib, M., Lin, C. T. & Blumenstein, M. A survey on object instance segmentation. *SN Comput. Sci.* https://doi.org/10.1007/s42979-022-01407-3 (2022).
43. Sun, R. et al. Survey of image edge detection. *Front. Signal Process.* **2**, 826967 (2022).
44. Zhang, Y. & Wang, P. A parallel thinning algorithm with two-subiteration that generates one-pixel-wide skeletons. In *Proceedings of 13th International Conference on Pattern Recognition*. Vol. 4. 457–461. https://doi.org/10.1109/ICPR.1996.547608 (1996).
45. Zhang, R., Isola, P., Efros, A. A., Shechtman, E. & Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 586–595 (2018).
46. Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint* arXiv:1609.02907 (2016).
47. Mikolov, T., Chen, K., Corrado, G. S. & Dean, J. Efficient estimation of word representations in vector space (2013).
48. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778 (2016).
49. Liu, Z. et al. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10012–10022 (2021).
50. Khanam, R. & Hussain, M. Yolov11: An overview of the key architectural enhancements. *arXiv preprint* arXiv:2410.17725 (2024).
51. Soria, X., Sappa, A., Humanante, P. & Akbarinia, A. Dense extreme inception network for edge detection. *Pattern Recognit.* **139**, 109461 (2023).
52. Cheng, B., Misra, I., Schwing, A. G., Kirillov, A. & Girdhar, R. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1290–1299 (2022).
53. Li, Q. et al. Boosting universal domain adaptation in remote sensing with dual-classifiers consistency discrimination and cross-domain feature mixup. *IEEE Trans. Geosci. Remote Sens.* **63**, 1–15. https://doi.org/10.1109/TGRS.2025.3571747 (2025).
54. Chen, W., Wen, Y., Zheng, J., Huang, J. & Fu, H. Ban: A universal paradigm for cross-scene classification under noisy annotations from rgb and hyperspectral remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **63**, 1–13. https://doi.org/10.1109/TGRS.2025.3539499 (2025).

Author contributions
Data collection and annotation was performed by Mai Viet-Hoang Do, Quang-Duy Pham, Van-Nam Hoang, Hoang Duy Linh Pham, Duc-Thang Phung. Mai Viet-Hoang Do and Thi-Lan Le wrote the first draft of the manuscript. All authors contributed to the state of the art analysis and commented on the manuscript. Thi-Lan Le had analyzed the results. Mai Viet-Hoang Do, Hoang Duy Linh Pham, Duc-Thang Phung have implemented the code. Van-Nam Hoang, Van-Sam Hoang, Michiel Vlaminck, Hiep Luong, Thanh-Hai Tran, Hai Vu critically revised the work. All authors read and approved the final manuscript.

Funding
This work is funded by VLIR SI project “DORA: open Drone data platfOrm for biodiveRsity mApping to support the sustainable management of forests”, under grant number VN2023SIN403A103. This research received partial funding from the Flemish Government under the 'Onderzoeksprogramma Artificiele Intelligentie (AI) Vlaanderen’ program.

Declarations

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to T.-L.L.