Communication Papers of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 1–8
DOI: 10.15439/2025F8182
ISSN 2300-5963 ACSIS, Vol. 45

Tree Segmentation from Low-Resolution Digital Orthophotos using a Hybrid Deep Learning Model

Irfan Abbas, Robertas Damaševičius, Rytis Maskeliūnas, Muhammad Abdullah Sarwar
Centre of Real Time Computer Systems, Kaunas University of Technology
Kaunas, Lithuania
irfan.abbas@ktu.edu, robertas.damasevicius@ktu.lt, rytis.maskeliunas@ktu.lt, m.sarwar@ktu.edu

***Abstract*—This study presents a cost-effective tree crown segmentation framework using a hybrid deep learning model that combines a ResNet-34 encoder with a U-Net decoder. Our approach operates on low-resolution RGB Digital Orthophotos (DOPs) collected from urban and peri-urban areas in Bochum, Germany, simulating real-world data constraints. We processed 450 orthophoto–mask pairs through a comprehensive preprocessing pipeline including resizing (from 20000×20000 to 256×256), augmentation, and noise simulation. The model was trained using 10-fold cross-validation, achieving a Dice coefficient of 0.8678, Intersection over Union (IoU) of 0.7754, precision of 0.8410, and recall of 0.9103. These results demonstrate that even with downsampled imagery, reliable segmentation of tree crowns is feasible, making our approach suitable for low-cost forest inventory and precision agroforestry applications. Unlike previous studies relying on high-resolution LiDAR, this work is among the first to show robust tree crown segmentation using low-resolution orthophotos, making it accessible for widespread use in resource-constrained settings.**

***Index Terms*—Tree segmentation, Digital Orthophotos, Remote Sensing, Forest Monitoring, Forest 4.0, Deep Learning.**

I. Introduction

FOREST monitoring is essential for ensuring the sustainable management, conservation, and restoration of forest ecosystems, which are critical to biodiversity, climate regulation, and human well-being [1]. By continuously tracking changes in forest cover, composition, and health, monitoring efforts support early detection of deforestation, forest degradation, pest outbreaks, and the impacts of climate change [2], [3]. With the increasing complexity of environmental challenges and the growing demand for data-driven decision-making, traditional forest monitoring methods are evolving toward more integrated, automated, and scalable solutions. This transformation is embodied in the concept of Digital Forestry, which leverages advanced technologies such as remote sensing, artificial intelligence, Internet of Things (IoT), and geospatial analytics to enhance forest observation and analysis. Within this context, the emergence of Forest 4.0 that integrates cyber-physical systems, real-time data processing, and predictive analytics to enable proactive decision-making, optimize resource use, and ensure ecological resilience [4].

This research paper has received funding from Horizon Europe Framework Programme (HORIZON), call Teaming for Excellence (HORIZON-WIDERA-2022-ACCESS-01-two-stage) - Creation of the centre of excellence in smart forestry "Forest 4.0" No. 101059985. This research has been co-funded by the European Union under the project "FOREST 4.0 - Ekscelencijos centras tvariai miško bioekonomikai vystyti" (Nr. 10-042-P-0002).

Recent advances in satellite imagery have created new opportunities for forest monitoring, including tree segmentation on a large scale [5], [6]. Laser imaging, detection, and ranging (LiDAR)-based high-resolution [7] satellite data can provide detailed information on forest structure, including tree height, tree trunk, and tree area; however, such data are not always available and can be costly [6], [8]. Low-resolution Red-Green-Blue (RGB) satellite imagery cannot provide detailed forest information, such as forest structure and the area covered by the trees [9]. So, it is very challenging to get details of tree structure from the low-resolution LiDAR-based RGB satellite data due to mixed pixels, blurry images, shadows, overlapping, and lack of spectral information [10]. Dataset preprocessing is required to get the forest and tree structure details from the RGB images [11]. These images estimate tree cover, species, biomass, changes in forest characteristics, tree shapes, wood calculations, trunk detections, and many others [12].

We propose a Deep Learning (DL)-based model for using low-resolution RGB satellite imagery from the German forest to segment the tree areas, which comprises of ResNet-34 and U-Net models. The contributions of this study are as follows:

- We introduce a custom hybrid model that integrates a ResNet34 encoder with a U-Net decoder, combining strong feature extraction capabilities with spatial reconstruction suited for semantic segmentation tasks on low-resolution imagery.
- The model is trained and validated on DOP and nDOM datasets from the German federal state of North Rhine-Westphalia, demonstrating the feasibility of using 256×256-pixel images for high-quality segmentation.
- A comprehensive data preprocessing framework is developed, including image resizing, augmentation, compression, and noise simulation, to enhance model robustness and generalization across heterogeneous landscapes.

II. Related Work

Multiple models have been developed to segment individual trees from forests and urban streets using high-resolution and low-resolution LiDAR point clouds and RGB images, and results have shown significant differences [12]. Few of these developed methods use to detect and segment the tree tops because these are the highest points in the RGB imagery and LiDAR point cloud data. By identifying and detecting the tree tops, the features of the trees are extracted and segmented. The

[13] develops a YOLO-based CCD-YOLO model to segment the individual tree using the LiDAR datasets collected from Beijing and Henan Polytechnic University, China.

Study [14] proposed a Fuzzy Center Segmentation (FCS) for ITS and used the ALS and TLS LiDAR-based datasets collected from the Jasper National Park, Montane Cordillera Ecozone, Canada, but did not segment the trees very well.

Study [15] used a watershed transformer based on U-net to perform ITS on high-resolution data sets collected from Bengaluru, India, and Gartow, Germany. The accuracy achieved by the author is 46.3% and the IoU is 71.2%. For the other dataset, the accuracy was 52% and the IoU was 72. 6%, which is better than on the other data set. The datasets used in this research are also high-resolution-based.

In [16], the author developed a nonparametric approach for ITS from LiDAR data, detecting dominant and co-dominant trees for 94% and 62% for intermediate dead and overtopped trees. The overall accuracy achieved by the author is 77% which is not good for the high-resolution dataset. The author proposed the 5-step approach to segment the trees, which is time-consuming and did not provide good results. The performance of this method was also affected by the complexity of the forest terrain and the conditions of the vegetation.

In [17], the author uses a method to automate tree segmentation for individual trees from complicated urban forests. The developed method did not provide good accuracy, and the accuracy matrices are not provided, but it recognizes the difficulties of the urban areas. The suggested accuracy is lower than the natural forest areas.

In [19], a custom Individual Tree Matching (ITM) algorithm was used to compare LiDAR-detected trees with 284 field-measured reference trees. For local maxima methods, a fixed 3 x 3m window applied to a non-smoothed canopy height model (CHM) achieved an F_1-score of 0.65 with 86% of trees are detected, while methods based on [21] achieved excellent crown segmentation with mean crown $radius < 0.5m$ of field-measured crown radius. For non-local maxima methods, the adaptive mean shift algorithm (AMS3D) performed well with F_1 score of 0.67 and a mean crown $radius < 0.1m$.

In [18], the author proposed an approach to extract, detect, and segment individual crowns using multispectral airborne LiDAR data. Trees crowns are initially segmented in the spatial domain using the mean shift algorithm, under-segmented crowns are identified using a Support Vector Machine (SVM) classifier and geometric features, and the crowns identified from classification are refined using mean shift in a joint feature space with spatial and multispectral data. The experiments on a total of ten forest plots in Ontario, Canada, quantify the differences in SVM's multispectral space data, and improve the detection rate of dominant trees from 82% to 88% while having better accuracy for detection in dense and clumped forests.

Table I highlights that while high-resolution LiDAR and UAV-based imagery dominate the field, their effectiveness is highly context-dependent. Methods such as the SVM combined with Mean Shift and LiDAR demonstrated strong performance in complex forest environments, particularly for clumped tree detection, achieving up to 84% segmentation accuracy. Techniques like Fuzzy Center Segmentation and the Watershed Transformer (U-Net) delivered inconsistent or suboptimal results despite using high-resolution data, pointing to limitations in handling diverse terrain and canopy structures. The Watershed Algorithm (WA), when applied to UAV LiDAR data in Eucalyptus plantations, achieved the highest F1-score (0.761), excelling particularly in low-density plots. This underscores the suitability of classical segmentation approaches in structured forest settings. Traditional methods such as the nonparametric 5-step approach, though reasonably accurate (77%), were found to be time-consuming and sensitive to terrain variations, limiting scalability. Urban ITS models consistently underperformed, highlighting persistent challenges in segmenting trees in built environments due to occlusions and background complexity. Deep learning-based methods like YOLO and U-Net offer promising segmentation capabilities but show mixed outcomes depending on dataset quality, annotation accuracy, and forest heterogeneity. The analysis reveals that no single approach outperforms others; instead, method effectiveness hinges on factors such as data resolution, forest density, canopy complexity, and the integration of geometric and machine learning strategies. These insights justify the development of hybrid models, such as the proposed U-Net with ResNet34 encoder, which aim to balance performance and generalizability, particularly when working with low-resolution RGB imagery in urban-natural mixed forest landscapes.

III. Methodology

We propose a hybrid deep learning model that combines the U-Net architecture with a ResNet34 encoder for tree segmentation from the German urban area. The model uses a U-Net architecture with the Resnet34 backbone and extracts features from the RGB satellite images. The proposed model use to train the dataset and applies the k-fold cross validation for 10 folds with the learning rate of le-2 and patience of 10. Each fold runs for 100 epochs, early stopping technique is used to stop the training on the best Dice score. The proposed model consists of the U-Net framework, Resnet34 encoder, Decoder with skip connections, and output layer.

A. U-Net Framework

U-Net is a fully convolutional neural network developed for pixel-wise segmentation tasks. A U-Net is structured in a symmetric "U" shape with a contracting path (an encoder) and an expanding or decoding path with skip connections between layers with the same index in the contraction and expansion paths. The contracting path extracts contextual information when downsampling the image and increasing the depth of features, whereas the expanding or decoding path extracts spatial information when upsampling the image.

B. ResNet34 Encoder

ResNet34 is a residual network with 34 layers that uses residual blocks with skip connections that help to smooth the

TABLE I
COMPARISON OF INDIVIDUAL TREE SEGMENTATION (ITS) METHODS

Ref.	Method/Model	Dataset Location/Type	Resolution	Accuracy/Performance	Notes
2009, [12]	Tree top detection with RGB/LiDAR	Forest & urban areas	High/Low	Varies significantly with resolution	Tree tops used for segmentation
2025, [13]	CCD-YOLO (YOLO-based)	Beijing & Henan Univ., China	High	Accuracy not specified	YOLO-based ITS with high-res LiDAR
x2024, [14]	Fuzzy Center Segmentation (FCS)	Jasper Nat. Park, Canada (ALS/TLS)	High	Poor segmentation	Ineffective for high-res data
2024, [15]	Watershed Transformer (U-net)	Bengaluru, India & Gartow, Germany	High	India: Acc. 46.3%, IoU 71.2%; Germany: Acc. 52%, IoU 72.6%	Better in Germany
2016, [16]	Nonparametric 5-step	Forest terrain	High	Overall 77%; Dom. 94%; Co-dom. 62%	Time-consuming; terrain sensitive
2016, [17]	Automated urban ITS	Urban forests	High	Low accuracy; metrics not provided	Urban segmentation issues noted
2025, [18]	SVM + Mean Shift + LiDAR	Ontario, Canada (10 plots)	High	Det. improved 82% $\rightarrow$ 88%, Segm. 84%	Effective for clumped trees; limited spectral use
2024, [19]	ITM, AMS3D, Local Maxima	Subtropical forests	High	Local Maxima: F1=0.65, Det.=86%; AMS3D: F1=0.67, radius error $<$ 0.1m	Limited by density (3 pts/m^2), GPS errors
2024, [20]	WA, LMA, EDCA, LSA (UAV LiDAR)	Eucalyptus plantations	High	WA: F1=0.761; CHM-based best	Sensitive to canopy/density; WA excels in low-density

gradient flow during training, as well as achieve a smooth gradient descent in smaller, deeper networks while avoiding the vanishing gradient dilemma in depth. The architecture allows for the ability to learn more complex features and tends to work better when lower-resolution imagery lacks fine detail. The model processes the input RGB images in different steps. In the first step, the model uses the 7x7 CNN layer, followed by batch normalization and the ReLU activation method. After that, a max pooling layer is used to minimize the spatial dimensions for all images. In the last four stages, residual blocks are applied to extract the high-level features. Every stage consists of multiple 3x3 convolutions to identify the shortcut, extract the best features, and repeat the process for the next stage.

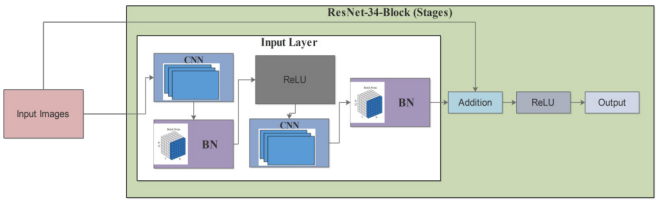

Fig. 1. The Input layer and stages of ResNet34 for Conv 7x7 layers, BatchNorm and ReLU

Figure 1 illustrates the architecture of a single ResNet-34 block, which serves as the encoder backbone within the hybrid deep learning model. The block operates on input RGB images and is composed of a series of convolutional, normalization, and activation layers structured around the concept of residual learning—a technique that enables the network to train effectively even as the number of layers increases, by preserving gradient flow and mitigating vanishing gradients. The process begins with the Input Layer, where the incoming image is passed through an initial Convolutional Neural Network (CNN) layer. This layer extracts low-level spatial features such as edges, color gradients, and textures. The resulting feature maps are passed through a Batch Normalization (BN) layer, which normalizes the outputs across the batch dimension. This step stabilizes and accelerates the training process by reducing internal covariate shift. Next, the normalized feature maps are passed through a ReLU (Rectified Linear Unit) activation function, which introduces non-linearity and allows the model to learn complex representations. The output from ReLU then flows into another CNN layer, which further refines the extracted features. Again, a batch normalization step follows to maintain consistent learning dynamics. ResNet has the shortcut connection (or identity mapping), which allows the input to bypass one or more convolutional layers and be directly added to the output of those layers. The original input image is routed directly to the Addition block, where it is combined with the output from the second batch normalization layer. This residual connection helps preserve the identity function and ensures that the model can learn effectively even when deeper layers are less informative. After addition, another ReLU activation is applied to the combined feature map, and the result is passed to the output of the ResNet block. This output can then be forwarded to the next block in the encoder or to downstream modules, depending on the architecture. Stacked ResNet blocks across multiple stages enable the encoder to extract increasingly abstract and hierarchical features, making it effective for complex tasks like tree crown segmentation from low-resolution satellite imagery. This structure allows the model to efficiently learn both fine details and broader spatial context, which is critical for accurately delineating individual trees in heterogeneous landscapes.

C. Decoder with Skip Connections

The decoder output of the U-Net reconstructs the spatial dimensions of the image progressively with the help of deconvolutions or upsampling layers. At each deconvolution step, the encoder feature maps are connected to the corresponding decoder layer with skip connections. These skip connections

help to safeguard fine-scale spatial information that may have been lost when downsampling, and they help the model demarcate individual tree tops, particularly when they are closely spaced together in dense forest areas. The decoder consists of different blocks, and each block consists of 2x2 transposed convolution, concatenation with the corresponding encoder, and two 3x3 convolutional layers followed by batch normalization and ReLU activation.

D. Output Layer

The last layer of the model is a 1x1 convolution to map from feature maps to a single-channel binary mask representing the probability that a pixel belongs to an individual tree crown. The output is produced using a sigmoid activation function, providing output values within the range [0,1]. During post-processing, the values can be threshold to get binary segmentation maps.

E. Full Architecture

Figure 2 illustrates the complete architecture of the proposed hybrid deep learning model for individual tree segmentation, which combines a ResNet-34-based encoder with a U-Net-style decoder. This encoder-decoder framework is specifically designed for semantic segmentation tasks using low-resolution RGB aerial images, and it extracts both spatial and contextual features to generate pixel-level binary masks of tree crowns.

The architecture begins with the Input Layer, which receives preprocessed aerial images of size 256×256 pixels. These images are passed into Encoder Path, where they are processed through four sequential ResNet34-Blocks (Stage-1 to Stage-4). Each block has multiple convolutional layers, batch normalization, ReLU activations, and residual connections, enabling the network to learn hierarchical features while maintaining gradient stability. As the data flows deeper through the encoder stages, the spatial dimensions are progressively reduced while feature depth increases, capturing increasingly abstract and high-level semantic information. Skip connections are established from each ResNet stage to its corresponding decoder block, preserving fine-grained spatial features that might otherwise be lost during downsampling.

Following the encoder, the data is passed into the Decoder Path, which consists of four Decoder Blocks arranged in reverse order to the encoder stages. Each decoder block performs a combination of upsampling (via transposed convolutions or bilinear interpolation), concatenation with the corresponding encoder features (via skip connections), and convolutional layers to refine the upsampled feature maps. This path helps reconstruct the original image resolution while selectively enhancing regions corresponding to individual tree crowns. The decoder progressively restores the spatial structure of the image, integrating detailed edge information with high-level semantic understanding.

At the final stage of the architecture, the processed feature map is passed through a Final Convolutional Layer followed by a Sigmoid Activation Function, which transforms the output into a binary probability map. Each pixel in this output represents the probability of belonging to a tree crown. A threshold (typically 0.5) is applied during post-processing to generate the final binary segmentation mask. These masks, as shown on the right side of the figure, effectively highlight tree structures within the urban or semi-natural landscape, with white or red representing detected tree areas and black for non-tree background.

This encoder-decoder model employs the powerful feature extraction capabilities of ResNet-34 alongside the spatial reconstruction strengths of U-Net, making it highly suitable for complex segmentation tasks on low-resolution imagery. The integration of skip connections is particularly important for maintaining localization accuracy in dense or heterogeneous forest and urban regions, leading to high-performance results across various segmentation metrics such as Dice coefficient, IoU, and precision-recall.

F. Performance Matrices

The following evaluation matrices are used to evaluate the performance of the proposed model:

The **Intersection-over-Union-(IoU)** is used for measuring the overlapping between mask images and the predicted mask:

$$\text{IoU} = \frac{|\text{Predicted Mask} \cap \text{Ground Truth (Mask)}|}{|\text{Predicted Mask} \cup \text{Ground Truth (Mask)}|} \quad (1)$$

where $\cap$ represents the intersection and $\cup$ represents the union between predicted segmentation and ground truth (Mask).

The **Dice Coefficient** is used for measuring the similarities between both predicted masks and ground truth:

$$\text{Dice} = \frac{2 \times |\text{Predicted Mask} \cap \text{Ground Truth}|}{|\text{Predicted Mask}| + |\text{Ground Truth}|} \quad (2)$$

where $\cap$ represents the intersection between the predicted segmentation and the ground truth.

Precision is used to compare the actual tree area and predicted tree area:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (3)$$

where TP (True Positives) represents the number of correctly predicted positive samples, and FP (False Positives) represents the number of incorrectly predicted positive samples.

Recall is used to evaluate the actual tree area that is correctly detected:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (4)$$

IV. Dataset and Setup

A. Data site

The data set is collected from the eastern area of Bochum, a city located in the federal state of North Rhine-Westphalia, Germany. Bochum is located in the Ruhr metropolitan area and has a strong industrial history and a diverse urban landscape. The eastern Bochum region, where the data is collected, consists of a wide range of residential areas, commercial areas, and greenery, providing a wonderful and representative sample of the region. The Ruhr area is highly urbanized but

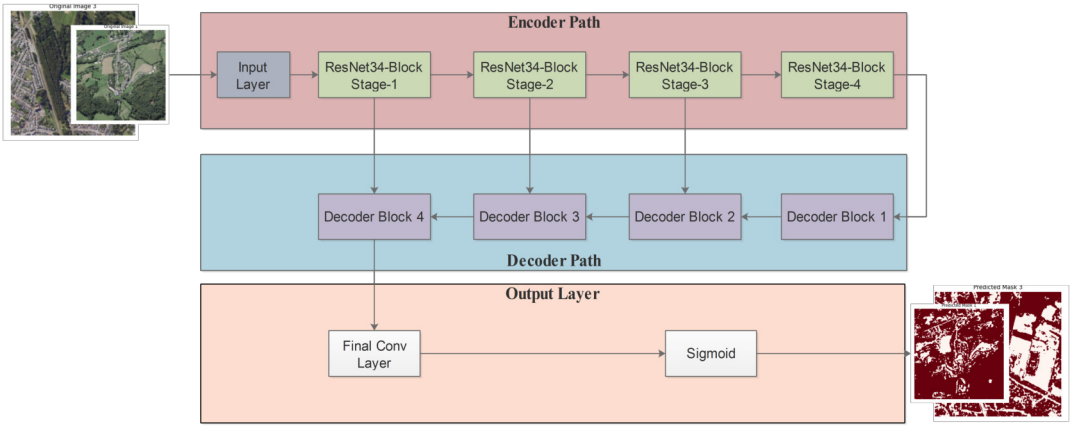

Fig. 2. A proposed hybrid model combines U-Net and Resnet34

quite dense. It consists of buildings, agricultural land, forests, and water areas. In recent decades, the region has changed significantly structurally. Heavy industries, such as coal and steel, have been replaced by other industries, causing serious changes in the landscape. This area has been affected by the closure of the Opel factory, which was driven by economic globalization in this city.

A study site is shown in Figure 3, which provides a multi-scale geographic overview of the study area, illustrating the location of Bochum city within the federal state of North Rhine-Westphalia, Germany. The rightmost panel (A) highlights North Rhine-Westphalia on the map of Germany, marking Bochum in red. Panel (B) shows a broader satellite view of the state with Bochum outlined, while panel (C) zooms into a high-resolution satellite image of Bochum, delineating its administrative boundaries in red. The study site is situated within the urban-natural landscape where the tree segmentation analysis was conducted.

B. Data Pre-processing

Data pre-processing is performed on 450 Digitale Orthophotos (DOP) images and 450 corresponding normiertes Digitales Orthophoto Modell (nDOM) images, collected from the GeoData Portal for the year 2023. Digitale Orthophotos is a German term meaning Digital Orthophotos, which are georeferenced aerial or satellite images that can be read and displayed using Geographic Information Systems (GIS) software. Both 450 DOP and 450 nDOM images are originally sized 20000x20000 pixels and converted into 256x256 pixels, with some noise added to create a blurring effect. An algorithm is proposed to convert all DOP and nDOM images into RGB format using Glymur and the Python Imaging Library (PIL/Pillow). The algorithm developed takes each image individually, processes it by converting it into Joint Photographic Experts Group (JPEG) format at 256x256 pixels, and stores it in a separate folder for further model implementation. After that, all DOPs and nDOMs are manually compared, and each DOP is renamed to match its corresponding nDOM for further tree area segmentation.

Figure 4 illustrates the data preprocessing workflow applied to DOPs and nDOMs, which are the primary input sources for the tree segmentation model. The preprocessing pipeline begins with the ingestion of raw high-resolution input images—each originally sized at 20,000 × 20,000 pixels. These inputs undergo several transformation steps to ensure compatibility with the deep learning model, improve computational efficiency, and simulate real-world data imperfections. The first stage, labeled ”Get the Input,” retrieves and parses the raw DOP and nDOM files, which are typically stored in JPEG 2000 (JP2) format. These files are then passed through a series of preprocessing operations. The raw images are resized from their original resolution to 256 × 256 pixels, significantly reducing memory load and speeding up training while maintaining sufficient detail for segmentation tasks. Additionally, image rotations are applied to introduce data augmentation, which enhances model generalization by exposing it to various spatial orientations of tree structures. Compression is another vital transformation step used to mimic the quality degradation commonly encountered in operational remote sensing data. This is followed by a format conversion process, where the images are transformed from JP2 to standard JPEG using the Python Imaging Library (PIL). PIL serves as the central processing engine in this pipeline, enabling all format handling, resizing, and augmentation operations. Once the core transformations are completed, the images are further processed by artificially introducing noise to simulate blurriness. This step is critical in emulating real-world conditions such as motion blur, atmospheric disturbances, or sensor imperfections, making the model more robust against such variations during inference. The final output of the pipeline consists of preprocessed JPEG and TIFF images that are uniformly scaled, augmented, and formatted. These prepared datasets serve as inputs to the hybrid deep learning segmentation model, enabling efficient training and evaluation under controlled yet realistic scenarios. The structured pipeline ensures data quality, uniformity, and robustness, all of which are crucial for achieving reliable performance in tree crown segmentation tasks using low-

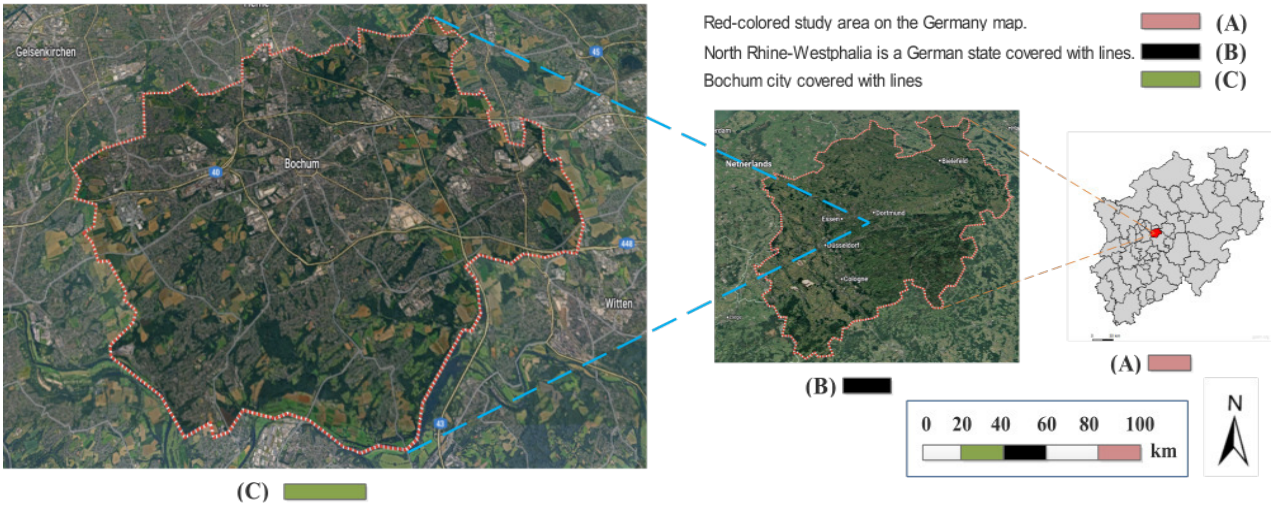

Fig. 3. (A): Indicate the study area in red color from the German Map, (B): Indicate the study area from the North Rhine-Westphalia state (C): The whole study area is located in the eastern part of Bochum

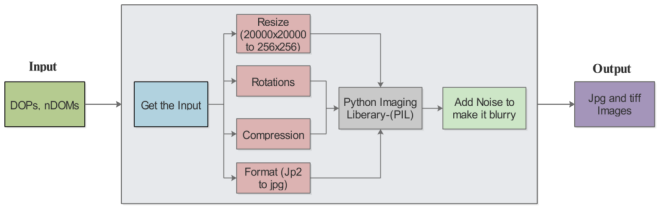

Fig. 4. Data pre-processing workflow using the PIL

TABLE III
THE SPECIFICATIONS OF EXPERIMENTAL SETUP

System Configuration	Hardware Specification
Programming Language	Python
Development Environment	Visual Studio
GPU	NVIDIA GEForce RTX 3080
CUDA Version	11.8
RAM	10 GB
Operating System	CentOS

resolution remote sensing imagery.

C. Experimental Setup

This experiment uses a DOPs- and nDOMs-based dataset for tree crown segmentation in a natural environment, aiming to evaluate the validity and effectiveness of the model's generalizability. The model was trained using hyperparameters given in Table II. The hardware specification for the experimental setup is shown in Table III

TABLE II
TRAINING HYPERPARAMETERS FOR MODEL DEVELOPMENT

Hyperparameter	Value
Learning Rate	1e-2 (with ReduceLROnPlateau)
Batch Size	16
Optimizer	Adam
Epochs	Up to 100 (early stopping on Dice score)
Validation Strategy	10-fold cross-validation

V. RESULTS AND DISCUSSIONS

This section presents and discusses the experimental results in detail. The data set is trained using the proposed model with 10-fold cross-validation. Each fold runs for up to 100 epochs, with early stopping applied to retain the best-performing model. The performance of the model is evaluated using key metrics: Dice coefficient, Intersection over Union (IoU), Accuracy, Precision, and Recall. The model achieves its best results with a Dice score of 0.8678, an IoU of 0.7754, an accuracy of 0.8180, a precision of 0.8410, and a recall of 0.9103. With comparison of baseline values, the proposed model consistently improves the results in training and in the validation process. The results show that the proposed model performs very well. Table IV summarizes the performance of cross-validation using the same matrices as the developed model. Using a 10-fold cross-validation approach, the model's ability to generalize across different subsets of the data set is evaluated. During the training, the dice-coefficient and IoU are the most critical metrics in this situation because they quantify spatial overlap between predicted tree regions and hand-annotated tree crowns. A high Dice value indicates that the model has accurately segmented the tree areas with little over- and/or under-segmentation. The best Dice Coefficient (0.8585) and IoU (0.7598) scored in Fold 10 suggests that this instance the model is able to produce such accurate tree segmentation despite the difficulties associated with low-resolution input data.

Accuracy measures the total number of correctly classified pixels (trees and background) as a proportion. Although accuracy is not important, it can be informative and less sensitive when using skewed data. The Fold 10 again has the highest

TABLE IV
THE TRAINING AND VALIDATION RESULTS FOR EACH FOLD

Folds	Dice Coefficient	IoU	Accuracy	Precision	Recall
Fold 1	0.7937	0.6613	0.7473	0.6928	0.9438
Fold 2	0.8133	0.7094	0.7577	**0.7938**	0.8949
Fold 3	**0.8445**	**0.7322**	**0.7820**	**0.7847**	0.9231
Fold 4	**0.8448**	**0.7383**	0.7729	0.7597	**0.9667**
Fold 5	0.7895	0.6799	0.7575	0.7716	0.8523
Fold 6	0.7895	0.6573	0.7260	0.6770	**0.9547**
Fold 7	0.8294	0.7102	0.7658	0.7560	0.9255
Fold 8	0.8363	0.7243	**0.7834**	0.7831	0.9118
Fold 9	0.7823	0.6459	0.7415	0.6736	**0.9417**
Fold 10	**0.8585**	**0.7598**	**0.8091**	**0.8150**	0.9169

accuracy (0.8091) supporting the model performance. The precision value measures the numbers of the predicted pixels for trees that were correct, so this is key in minimizing false positives, for example, with roads or shadows on trees. The model also achieved its highest precision (0.8150) in Fold 10, demonstrating that it accurately differentiates trees from other land features: houses, land area, roads, and other objects.

In recall, the false negative refers to the correctly detected number of tree pixels, as this is relevant in ecological or forestry applications when missing tree areas could affect the estimation of canopy coverage. Fold 4 had the best recall (0.9667), which means that the model detected nearly every actual tree pixel in Fold 4, although it was possible to add false positives. Although Fold 10 consistently produced the highest in all metrics used in this study, Fold 3 and Fold 4 consistently perform well after Fold 10 and gain the second highest results. Fold 9 had the lowest results for Dice (0.7823) and IoU (0.6459); it is probable that Fold 9 had more challenging image conditions, such as shadowing, overlapping trees, and/or poorer annotation quality. In the same vein, precision was at its lowest in Fold 9 (0.6736), which may indicate more confusion with non-tree elements.

The proposed hybrid model demonstrates solid and steady performance through every fold, with some minor variations that may be explained by the natural variance of low-resolution orthophoto imagery. The consistently high Dice and IoU scores indicate that the hybrid deep learning model is well-tailored for tree crown segmentation tasks with the inevitable constraints of resolution and noise in the input data. These results underscore the viability of deep learning models, particularly ResNet34 with a U-Net decoder, to use low-resolution, remote sensing data to extract important spatial features for forestry and ecological monitoring.

We tested our part of the dataset on the trained model and visualized the results including the original image, mask image, predicted mask, and overlapped image. The results for each sample are also measured with the in performance matrices. The results for the 15 samples are given in Table V. The model performs very well on sample 8 and achieved the highest Dice coefficient of 0.8995, IoU of 0.8174, accuracy of 0.8558, and precision of 0.8655, but sample 5 achieves the highest recall. During testing, the second sample produces the second largest results with a Dice coefficient of 0.8812, and an IoU of 0.7877.

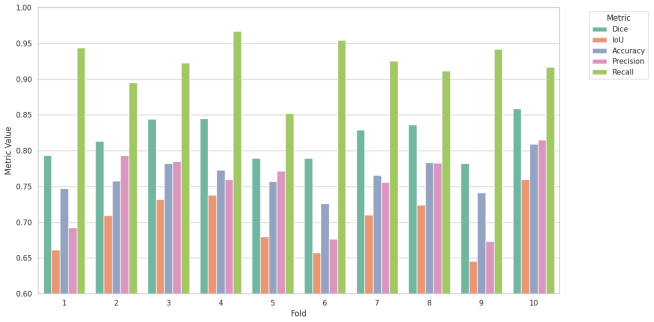

Fig. 5. Borplot visualization of model training and validation segmentation performance Metrics (Dice Coefficient, IoU, Accuracy, Precision, and Recall) across 10 folds

TABLE V
THE TESTING RESULTS FOR 15 DIFFERENT SAMPLES

Samples	Dice Coefficient	IoU	Accuracy	Precision	Recall
Sample 1	0.7220	0.5650	0.6410	0.5681	0.9903
Sample 2	0.6393	0.4693	0.5780	0.4732	0.9850
Sample 3	0.7328	0.5783	0.5942	0.5860	0.9777
Sample 4	0.7268	0.5709	0.6609	0.5798	0.9854
Sample 5	**0.8812**	**0.7877**	0.8009	0.7903	**0.9958**
Sample 6	0.7162	0.5579	0.5734	0.5594	**0.9952**
Sample 7	0.8328	0.7135	0.7645	0.7638	0.9155
Sample 8	**0.8995**	**0.8174**	**0.8558**	**0.8655**	0.9362
Sample 9	0.8600	0.7544	**0.8464**	**0.8052**	0.9228
Sample 10	0.7705	0.6266	0.7006	0.6332	0.9837
Sample 11	0.7901	0.6531	0.6609	0.6571	0.9906
Sample 12	0.7707	0.6269	0.6332	0.6325	0.9862
Sample 13	0.8588	0.7525	0.7890	0.7600	0.9871
Sample 14	0.8047	0.6733	0.7294	0.6836	0.9780
Sample 15	0.7380	0.5833	0.6977	0.5867	0.9900

The results are summarized visually in Figure 6. The model achieves high recall, meaning it is very sensitive to identifying tree regions, though at the cost of reduced precision. The Dice Coefficient and Accuracy metrics confirm its overall reliability in segmentation tasks, while the IoU values reflect room for improvement in terms of spatial precision, especially under complex visual conditions. These results validate the effectiveness of the hybrid ResNet34–U-Net architecture in tree segmentation from low-resolution aerial images, while also suggesting directions for refinement—particularly in improving boundary sharpness and reducing false positives.

VI. CONCLUSIONS

We proposed a hybrid deep learning model that combines the ResNet34 encoder with a U-Net decoder architecture to segment individual trees from low-resolution RGB orthophoto images (DOPs) over the German urban and semi-urban landscape. Despite the limitations of low-resolution imagery, the model demonstrated high segmentation performance, achieving a Dice coefficient of 0.8678, IoU of 0.7754, accuracy of 0.8180, precision of 0.8410, and an exceptional recall of

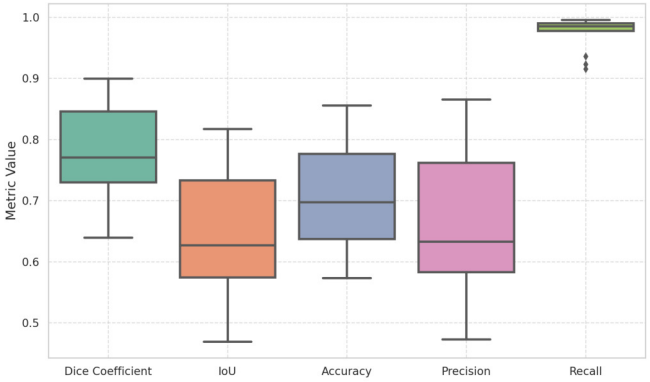

Fig. 6. Boxplot visualization of segmentation performance metrics (Dice Coefficient, IoU, Accuracy, Precision, and Recall)

0.9103. These results confirm the model's strong ability to detect and delineate tree crowns across diverse environments, including dense forest patches and complex urban settings. The preprocessing pipeline—consisting of format conversion, resizing, augmentation, and noise injection—played a critical role in preparing the dataset and enhancing model generalizability. Visualizations of predicted masks and overlay comparisons further validated the model's effectiveness, showing a high degree of alignment with ground truth annotations, particularly in structured and less cluttered regions.

Comparative analysis with existing state-of-the-art individual tree segmentation (ITS) methods revealed that our approach is competitive even against models relying on high-resolution LiDAR or UAV data, making it a cost-effective alternative for large-scale forest monitoring in data-limited regions. The model performed well overall, but challenges remain in improving segmentation precision and handling complex urban-object boundaries, suggesting opportunities for future work on post-processing refinement and attention-based enhancements.

Data Availability

The data set is collected from the GeoData Portal of the Federal State of North Rhine-Westphalia under the data license "Deutschland - Zero - Version 2.0" for the year 2023.

References

[1] D. Rajasugunasekar, A. K. Patel, K. B. Devi, A. Singh, P. Selvam, and A. Chandra, "An integrative review for the role of forests in combating climate change and promoting sustainable development," *International Journal of Environment and Climate Change*, 2023.

[2] R. Damaševičius and R. Maskeliūnas, "Modeling forest regeneration dynamics: Estimating regeneration, growth, and mortality rates in lithuanian forests," *Forests*, vol. 16, no. 2, 2025.

[3] ——, "Adaptive sensor clustering for environmental monitoring in dynamic forest ecosystems," *Peer-to-Peer Networking and Applications*, vol. 18, no. 3, 2025.

[4] R. Damaševičius, G. Mozgeris, A. Kurti, and R. Maskeliūnas, "Digital transformation of the future of forestry: an exploration of key concepts in the principles behind forest 4.0," *Frontiers in Forests and Global Change*, vol. 7, 2024.

[5] T. Mijit, E. Firkat, X. Yuan, Y. Liang, J. Zhu, and A. Hamdulla, "Lr-seg: A ground segmentation method for low-resolution lidar point clouds," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 347–356, 2024.

[6] D. Joshi and C. Witharana, "Vision transformer based unhealthy tree crown detection and evaluation of annotation uncertainty," 2025.

[7] L. Wallace, A. Lucieer, and C. S. Watson, "Evaluating tree detection and segmentation routines on very high resolution uav lidar data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 12, pp. 7619–7628, 2014.

[8] M. D. Hossain and D. Chen, "Remote sensing image segmentation: Methods, approaches, and advances," *Remote Sensing Handbook, Volume II*, pp. 117–144, 2025.

[9] H. Chen, W. Li, J. Gu, J. Ren, H. Sun, X. Zou, Z. Zhang, Y. Yan, and L. Zhu, "Low-res leads the way: Improving generalization for super-resolution by self-supervised learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 25 857–25 867.

[10] B. Koonce, "Resnet 34," in *Convolutional neural networks with swift for tensorflow: image recognition and dataset categorization*. Springer, 2021, pp. 51–61.

[11] J. Shen, Q. Xu, M. Gao, J. Ning, X. Jiang, and M. Gao, "Aerial image segmentation of nematode-affected pine trees with u-net convolutional neural network," *Applied Sciences*, vol. 14, no. 12, p. 5087, 2024.

[12] J. Reitberger, C. Schnörr, P. Krzystek, and U. Stilla, "3d segmentation of single trees exploiting full waveform lidar data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 64, no. 6, pp. 561–574, 2009.

[13] Y. Liu, A. Zhang, and P. Gao, "From crown detection to boundary segmentation: Advancing forest analytics with enhanced yolo model and airborne lidar point clouds," *Forests*, vol. 16, no. 2, p. 248, Jan. 2025.

[14] Z. Xi, C. Hopkinson, and L. Chasmer, "Supervised terrestrial to airborne laser scanner model calibration for 3d individual-tree attribute mapping using deep neural networks," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 209, pp. 324–343, 2024.

[15] M. Freudenberg, P. Magdon, and N. Nölke, "Individual tree crown delineation in high-resolution remote sensing images based on u-net," *Neural Computing and Applications*, vol. 34, no. 24, p. 22197–22207, Aug. 2022.

[16] H. Hamraz, M. A. Contreras, and J. Zhang, "A robust approach for tree segmentation in deciduous forests using small-footprint airborne lidar data," *International journal of applied earth observation and geoinformation*, vol. 52, pp. 532–541, 2016.

[17] C. Zhang, Y. Zhou, and F. Qiu, "Individual tree segmentation from lidar point clouds for urban forest inventory," *Remote Sensing*, vol. 7, no. 6, pp. 7892–7913, 2015.

[18] S. Li, S. Zhao, Z. Tian, H. Tang, and Z. Su, "Individual tree segmentation based on region-growing and density-guided canopy 3d morphology detection using uav lidar data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2025.

[19] T. Saeed, E. Hussain, S. Ullah, J. Iqbal, S. Atif, and M. Yousaf, "Performance evaluation of individual tree detection and segmentation algorithms using als data in chir pine (pinus roxburghii) forest," *Remote Sensing Applications: Society and Environment*, vol. 34, p. 101178, 2024.

[20] Y. Yan, J. Lei, J. Jin, S. Shi, and Y. Huang, "Unmanned aerial vehicle–light detection and ranging-based individual tree segmentation in eucalyptus spp. forests: Performance and sensitivity," *Forests*, vol. 15, no. 1, p. 209, 2024.

[21] M. Dalponte and D. A. Coomes, "Tree-centric mapping of forest carbon density from airborne laser scanning and hyperspectral data," *Methods in ecology and evolution*, vol. 7, no. 10, pp. 1236–1245, 2016.