Neural Computing and Applications (2022) 34:22197–22207
https://doi.org/10.1007/s00521-022-07640-4

ORIGINAL ARTICLE

Individual tree crown delineation in high-resolution remote sensing images based on U-Net

Maximilian Freudenberg[1] · Paul Magdon[2] · Nils Nölke[1]

Received: 22 December 2021 / Accepted: 18 July 2022 / Published online: 16 August 2022

Abstract

We present a deep learning-based framework for individual tree crown delineation in aerial and satellite images. This is an important task, e.g., for forest yield or carbon stock estimation. In contrast to earlier work, the presented method creates irregular polygons instead of bounding boxes and also provides a tree cover mask for areas that are not separable. Furthermore, it is trainable with low amounts of training data and does not need 3D height information from, e.g., laser sensors. We tested the approach in two scenarios: (1) with 30 cm WorldView-3 satellite imagery from an urban region in Bengaluru, India, and (2) with 5 cm aerial imagery of a densely forested area near Gartow, Germany. The intersection over union between the reference and predicted tree cover mask is 71.2% for the satellite imagery and 81.9% for the aerial images. On the polygon level, the method reaches an accuracy of 46.3% and a recall of 63.7% in the satellite images and an accuracy of 52% and recall of 66.2% in the aerial images, which is comparable to previous works that only predicted bounding boxes. Depending on the image resolution, limitations to separate individual tree crowns occur in situations where trees are hardly separable even for human image interpreters (e.g., homogeneous canopies, very small trees). The results indicate that the presented approach can efficiently delineate individual tree crowns in high-resolution optical images. Given the high availability of such imagery, the framework provides a powerful tool for tree monitoring. The source code and pretrained weights are publicly available at https://github.com/AWF-GAUG/TreeCrownDelineation.

Keywords Deep learning · U-Net · Remote sensing · Tree · Delineation · Segmentation

1 Introduction

The size and structure of a tree crown determines the primary production of a tree and is characterized by the species-specific branching pattern, site conditions, and resource competition, mainly for light. One of the most important tree crown characteristics is the crown projection area (CPA), which can be defined as the parallel vertical projection of the tree crown onto a horizontal plane. Delineating tree crowns to derive the CPA can provide valuable information at the single tree and stand level. In forest management, the single tree level CPA is used to predict diameter, volume [1], and growth rates [2] of individual trees. Tree crown maps generated at the stand level can be utilized to model stand competition [3] and to study canopy gap patterns [4].

Producing tree crown maps by manual delineation and visual interpretation of aerial images has a long history in forestry and ecology. However, the manual delineation of tree crowns is laborious and time-consuming and is often only practical for small areas. Therefore, automatic delineation methods have been developed that have the potential to map a large number of tree crowns with lower effort. Methods for individual tree crown delineation (ITD) in high resolution optical images include watershed segmentation [5], template matching [6], multi-scale segmentation [7] and region growing approaches [8]—a review is

✉ Maximilian Freudenberg
maximilian.freudenberg@uni-goettingen.de

Paul Magdon
paul.magdon@hawk.de

Nils Nölke
nnoelke@gwdg.de

[1] Forest Inventory and Remote Sensing, University of Göttingen, Büsgenweg 5, 37077 Göttingen, Germany

[2] Faculty of Resource Management, HAWK Göttingen, Büsgenweg 1a, 37077, Göttingen, Germany

provided in [9]. These methods have been successfully applied at the stand or district level, but it is difficult to transfer them to larger areas. This is due to their inherent sensitivity to the image quality (e.g., spatial resolution, light conditions, phenology), which results in poor performance when applied to heterogeneous large area mappings.

With the availability of 3D data observed with light detection and ranging (LiDAR) sensors or created by photogrammetric image analysis, new approaches based on either 2.5D surface models or directly working on 3D point clouds have become available. An overview of these approaches is presented in [10]. However, such 3D datasets are costly to collect and thus are often not available for large areas or regular time intervals, which limits the possibilities of applying tree crown delineation to larger regions. In comparison, 2D images with high spatial resolution can be collected at much lower costs and higher frequency using satellites, aircrafts or unmanned aerial vehicles (UAVs). Consequently, a robust, fast, and automatic method for tree crown delineation based on 2D optical imagery is needed. To summarize, our study is motivated by: (1) The need for accurate data for forest monitoring, (2) the insufficient performance of existing methods, and (3) the desire to avoid costly LiDAR data.

To meet these challenges, we developed a new deep learning-based instance segmentation method for automatically delineating tree crown polygons in optical images with high spatial resolution. Compared to other methods, such as Mask-R-CNN [11], our method requires less training data, which renders it applicable to small study areas. To test and demonstrate the robustness of the proposed method, it is applied to two case studies from different environments (i.e., an urban area in Bangalore, India, and a forested area in Gartow, Germany) and with two different sensor types (WorldView-3 satellite and aerial images).

2 Related work

Recent advances in the field of deep learning provide new image analysis methods that are able to solve previously impossible computer vision tasks. For example, accurate object detection is possible under challenging conditions like varying illumination, object size, or viewing angle [11–13]. Image segmentation (i.e., pixelwise classification) networks are particularly interesting in the field of remote sensing, e.g., for wildfire detection [14], change detection [15] or landcover classification [16]. Numerous works have applied deep learning for the classification of tree species in remote sensing data, for example in Sentinel-2 satellite time series [17], in UAV imagery [18] or a combination of LiDAR data and satellite images [19].

A number of recent studies present applications of deep learning methods for individual tree detection, but many of them focus only on small areas or homogeneous plantations with disjunct tree crowns [20, 21]. A recent study by Weinstein et al. [22, 23] went further and trained a neural network for predicting bounding boxes around trees from aerial imagery with a spatial resolution of 10 cm on large areas distributed across the USA. They used a RetinaNet [24] architecture and reached a precision of 61% and a recall of 69%. The bounding boxes alone already provide valuable information, e.g., on tree density—but it is insufficient for many of the described applications where individual tree crown polygons are needed.

A study by Braga et al. [25] used the Mask-R-CNN [11] network to delineate tree crowns for a small study area in Brazil, using WorldView-2 satellite imagery with 0.5 m spatial resolution. They reported an average bounding box precision of 91% and a recall of 81%. The drawback of Mask-R-CNN is the need for large amounts of labeled training data with individual tree crowns, which are often not available. To overcome the training data limitation, they generated synthetic training samples from a limited number of manually digitized tree crowns. Since they tested the approach in a single study area only, it remains unclear how well the method can be transferred to other areas.

3 Methods

3.1 Model

To delineate the tree crowns, a two-step approach was applied: First, a neural network predicted a tree cover mask, crown outlines, and the distance transform for each tree. Then, in a polygon extraction step, a conventional watershed transform was applied to a modified distance transform (see Eq. (1)), thereby extracting individual crowns. This process was inspired by TernausNetV2 [12] and DeepWatershed [26]. The key idea was to enhance the concept of TernausNetV2 (predicting object outlines) by feeding its output to a simplified DeepWatershed network, thereby creating a network output that is more suitable to be processed by conventional watershed algorithms.

Step 1—Tree crown mask generation: We employed a neural network, which encompassed two subnetworks (see Fig. 1): The first subnetwork generated a tree cover mask and a prediction for tree crown outlines based on the input imagery, analogous to [12]. This output was concatenated with the original input image and fed into the second

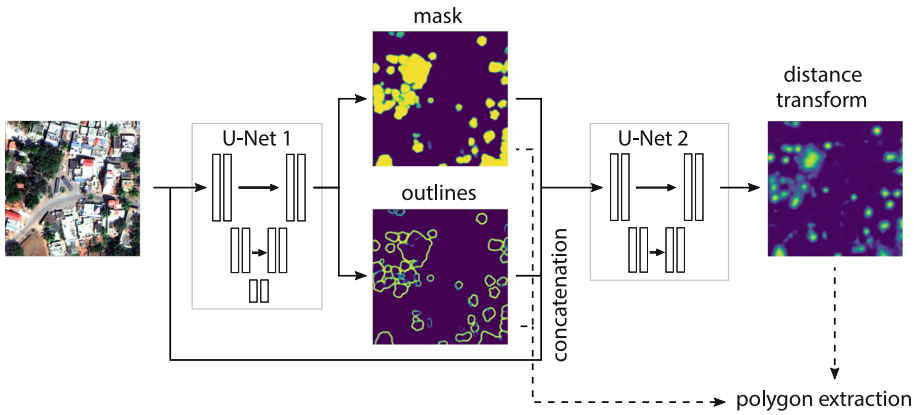

Fig. 1 The crown mask generation model in step 1 consisted of two subnetworks; a deep and a shallow U-Net with ResNet18 backbone (displayed only schematically), having five and three pooling stages, respectively. The first generated a tree mask and outlines, and the second predicted the distance transform. Mask, outlines, and distance transform were then used to obtain the final tree crown polygons in a subsequent polygon extraction step via watershed transform

subnetwork, which in turn output a distance transform. The distance transform measures how far a given tree crown pixel (foreground) is from the closest non-tree crown pixel (background); it is higher at the tree crown center and lower at the edges. Therefore, local maxima in the distance transform served as a proxy to the stem position. Finally, the full model output the tree cover mask, the outline prediction, and the distance transform.

Subnetwork one is a U-Net [27] with ResNet18 [28] backbone and five pooling stages, as provided by [29]. The second subnetwork had the same architecture and backbone, but only three pooling stages.

Step 2—Tree crown polygon extraction: Based on the model outputs, a series of processing steps is conducted to obtain the final tree crown polygons. Fig. 2 depicts the procedure. First, the mask, outline, and distance transform were combined according to the following formula

$$R\left(M,O,D\right)=\Theta\left(M^{\alpha}-\beta\,O^{\gamma}\right)D^{\delta} \tag{1}$$

where R is the transformed tree mask, Θ is the Heaviside function, M is the tree cover mask output by the network, O the outlines, and D the distance transform. Alpha, beta, and gamma are model parameters that were found via a hyperparameter tuning process using Hyperopt [30]. We found $\alpha=2$, $\beta=5$, $\gamma=1$ and $\delta=0.5$ to be a good choice and therefore conducted all experiments with this set. The search procedure is described in Sect. 4. The involved subtraction, exponentiation, and multiplication were applied pixel-wise.

The resulting transformed tree mask R was blurred using a Gaussian filter with standard deviation σ, followed by a local maximum search. The library in use [31] requires a minimum distance $d_{\min}$ and minimum height of the maxima $h_{\min}$, which can be used to tune the results. These maxima were finally used as a seed for the morphological watershed segmentation (implemented in [31]), which we restricted to areas where R exceeded a certain threshold t, which was also a hyper-parameter. To remove unrealistically small tree crowns, the resulting polygons were filtered by their size using a fixed threshold of 3 m^2.

3.2 Loss functions

The loss function for training the mask and outline detection network was a combination of binary cross-entropy (BCE) loss with the negative log of the intersection over union (IoU, for a definition see 3.3).

$$L_{\text{mask/outline}}(y_p,y_t)=\frac{1}{N}\sum_{i=1}^{N}\text{BCE}\left(y_{p,i},y_{t,i}\right)-\log(\text{IoU}(y_p,y_t)) \tag{2}$$

Where y_p and y_t are the predicted and ground truth masks/outlines and N is the number of pixels in the image, and i is the pixel index. For individual pretraining of the first

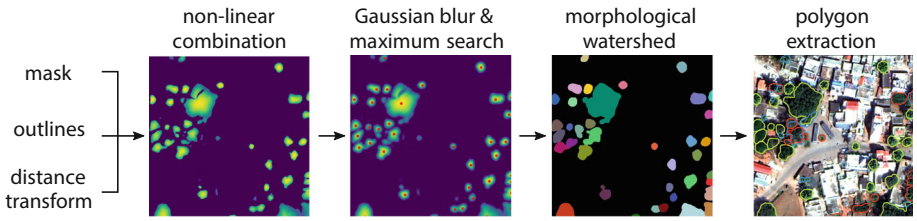

Fig. 2 In step 2, the tree crown polygons were extracted in four stages

subnetwork, the loss was calculated separately for masks and outlines and then summed up.

The second subnetwork was trained using the mean squared error between the predicted and ground truth distance transform.

$$L_{\text{dist}}(y_p, y_t) = \frac{1}{N}\sum_{i=1}^{N}(y_{p,i} - y_{t,i})^2 \tag{3}$$

The reference data for the distance transform were generated on a per-object basis and then normalized; the point in an object with the highest distance to the background had a value of one, linearly falling off to zero at the object edges. The classification into distance classes as performed in [26] was neglected, and the final loss function L for end-to-end training of the model was simply the sum of all three auxiliary losses:

$$L = L_{\text{mask}} + L_{\text{outline}} + L_{\text{dist}} \tag{4}$$

3.3 Performance metrics and evaluation

We calculate accuracy, precision, and recall for the polygon predictions based on the counts of true-positive, false-positive, and false-negative polygons (see Appendix). Furthermore, the intersection over union (IoU) of the tree cover mask is reported. The intersection over union can be best described using a ground truth vector $\vec{y}_t$ and prediction vector $\vec{y}_p$, which can be obtained by simply stacking the columns of the target/prediction matrix. Polygons with an IoU greater than 0.5 were counted as true positives.

$$\text{IoU}(\vec{y}_t, \vec{y}_p) = \frac{\vec{y}_t \cdot \vec{y}_p}{\|\vec{y}_t\| + \|\vec{y}_p\| - \vec{y}_t \cdot \vec{y}_p} \tag{5}$$

$\vec{y}$: Vector containing the ground truth or prediction
$\| \ldots \|$: Sum of all vector elements

In addition to the above metrics, which are based on true and false polygon counts, we report the pixel-wise precision, recall and IoU averaged over all pairs of correctly detected polygons and corresponding references.

To analyze whether our method tends to over-segment, we define the over-segmentation ratio O as follows:

$$O(\underline{t}, \underline{p}) = \frac{1}{N_t}\sum_{i,j}\Theta\left(\frac{\|t_i \cap p_j\|}{\|p_j\|} - 0.5\right) \tag{6}$$

$\underline{t}, \underline{p}$: List containing true or predicted polygons, N_t: Number of ground truth polygons, $\| \ldots \|$: Polygon area

The over-segmentation ratio counts the average number of predicted polygons which overlap with more than 50% of their area with a ground truth polygon. It makes no statement regarding segmentation quality.

4 Experimental setup

The performance of the new tree crown delineation method was assessed under varying site conditions and with different image sources. To test the robustness, we selected two contrasting scenarios: (1) delineation of urban trees located in the megacity of Bengaluru, India, using satellite images, and (2) delineation of trees in a forest area in Lower Saxony, Germany, for which aerial imagery was used.

4.1 Scenario 1: satellite Imagery

Image data: In the first experiment, urban trees in Bengaluru, India, were delineated using satellite imagery with 30 cm spatial resolution (after pan-sharpening) and eight VNIR bands. The Bengaluru image dataset covers an area of 5×50 km (∼12.94° N 77.56° E to 13.39° N 77.61° E) and was acquired on 2016-11-16 by WorldView-3 (DigitalGlobe ®) under cloud-free conditions. For all images, we added the normalized difference vegetation index (NDVI) to the image stack. Pansharpening was performed using the algorithm implemented in PCI Geomatica 2020 with default settings. The imagery underwent no atmospheric correction.

Reference data preparation: For training and validation of urban trees within the Bengaluru image dataset, we manually delineated all trees on screen in 35 tiles of 9 ha each by means of visual interpretation. These tiles were randomly split into 28 images for training and 7 images for validation. The tile locations were manually chosen to cover the landscape variety and number of trees within it. As not all trees were visually separable (e.g., in closed, homogeneous canopies), we labeled non-separable crown covered areas with contiguous polygons as tree groups. To perform independent tests, 23 additional one-hectare plots were used, for which the outline of a tree crown was drawn on-site using printed plot maps and in a second step digitized as a polygon. The in situ reference polygons were rasterized to match the extent and resolution of the satellite imagery. The outlines of these polygons were rasterized as well and dilated (widened) by two pixels to provide a stronger training signal. To obtain the in situ reference data for the distance transform, each tree crown polygon was rasterized, i.e., being one within the polygon and zero in the background. Then, each polygon was distance transformed; each pixel then represented the distance to the closest background pixel. Lastly, we normalized the distance transform to its maximum on a per-polygon basis, so that for all polygons, independent of size, the point farthest from the background obtains a value of one, linearly falling off to the edges.

Before we commenced training on the aforementioned data, we pretrained the neural network using an additional, separate pretraining dataset containing 330 one-hectare tiles with tree cover labels but without individual tree annotation.

Training procedure: In the pretraining step, we trained the first subnetwork on 264 tiles of one ha each for 200 epochs with an initial learning rate of $3 \cdot 10^{-4}$ and a batch size of 16. For validation, the remaining 66 images (20%) were used. The learning rate was periodically decreased according to a cosine annealing learning rate schedule [32] with a period length of 30, which was doubled after each period. As a loss function, the segmentation loss described in Eq. (2) was used. The pretraining aimed at achieving good tree/non-tree separation before learning individual tree separation.

In a second step, we trained the full model. For this task, we first determined the number of training epochs, maximizing model performance while avoiding overfitting, on 28 randomly selected tiles out of the 35, for which we had delineated individual trees. Then, we carried out training on the whole training dataset for the determined number of 89 epochs with the same learning rate schedule and settings as the pretraining. In addition to the segmentation loss, we penalized the mean squared error of the distance transform prediction, according to eq. (4). During training, the input images were augmented using fixed size random crops of 256×256 pixels, random vertical flips and 90° rotations.

Polygon extraction: Table 1 lists all polygon extraction parameters. Following [33], we set a minimum area of 3 m^2 to suppress polygons that were too small. The other parameters were optimized in a parameter search procedure on the validation dataset using [30]. We utilized the overall accuracy as an optimization goal and sampled the variables from uniform distributions using the "Tree of Parzen Estimators" (TPE). The distribution ranges are given in Table 1. Optimization was carried out for 200 iterations.

Evaluation: We applied the model to our test set of 23 images of 1 ha each and calculated the intersection over union between predicted and ground truth polygons. Predicted polygons with an IoU of more than 50% with any ground truth polygon were counted as true positives and all others as false positives. Ground truth polygons that had no matching prediction were consequently false negatives. False positives and negatives could occur even if the network correctly detects pixels as "tree" but failed to partition them correctly into their respective tree crowns or crown groups.

4.2 Scenario 2: aerial imagery

Image data: In the second experiment, we delineated trees in a dense forest area in Gartow, Germany, located near 52.98° N 11.42° E. The site covers an area of ca. 142 km^2 and the forest is mainly composed of Scots Pine (*Pinus sylvestris*) as the dominating species and other species like spruce (*Picea abies*), larch (*Larix spp.*), beech (*Fagus sylvatica*), and oak (*Quercus spp.*) are interspersed. The aerial images used were collected in 2018 using an Ultra Cam Falcon f100 digital camera and a LiteMapper 700 Laser Scanner with a nominal point density of $>10\,\mathrm{pts}/\mathrm{m}^2$. All images were georeferenced using the Global Naviagation Satellite System (GNSS) and Inertial Measurement Unit (IMU) sensor data of the platform, as well as ground control points (GCP) collected in the study area. The images were radiometrically calibrated and comprise the red, green, blue and near infrared bands. Again, we appended the NDVI to the image stack. For the orthorectification, a digital surface model was derived from the filtered LiDAR point cloud, which resulted in a true orthomosaic with a spatial resolution of 5 cm. Ortho-image generation was done using the photogrammetry tools of the PCI Geomatica software. Additionaly, we used the LiDAR data to create a canopy height model (CHM) with a spatial resolution of 0.25 m. The latter was *only* used to support the visual delineation of tree crowns for the reference data.

Reference data preparation: In total, 3674 trees of various species were labeled in 39 plots of 50 x 50 m. Within these plots, all separable tree crowns were delineated and checked against a canopy height model. Non-separable tree crowns were labeled as well and included in the tree cover mask, but no outlines or distance transforms

Table 1 Polygon extraction parameters used in the two scenarios of the experiment

		Satellite		Aerial	
Param	Description	value	Search space	value	Search space
$d_{\min}$	Minimum distance	3 m	1–10 m	2 m	1–5 m
$a_{\min}$	Minimum area	3 m^2	–	3 m^2	–
$h_{\min}$	Peak min. height	0.5	0.01–0.99	0.1	0.01–0.99
t	Mask threshold	0.1	0.01–0.99	0.1	0.01–0.99
σ	Gauss filter std. dev.	2 px	0–5 px	6 px	0–20 px

were calculated. The vector data were rasterized analogous to the procedure for case study one—with one exception: The outlines were dilated by four instead of two pixels due to the higher spatial resolution of the images. Table 2 gives an overview of the dataset properties.

Training procedure: In absence of an independent test dataset, a tenfold cross-validation on the 39 training plots was performed. Thirty-two images were used for training and seven for validation. In each cross-validation run, the network was trained for 89 epochs with the same settings as in the first case study but an initial learning rate of $5 \cdot 10^{-4}$.

Polygon extraction: To account for the higher spatial resolution and smaller size of the trees, the search space of the parameter optimization was adjusted; we decreased the minimum tree distance as the trees in Gartow tend to be smaller and increased the standard deviation of the Gaussian filter, as the resolution was higher. Since we used cross-validation on this dataset, the optimization was carried out on the training data for 200 steps.

Evaluation: For each cross-validation run, we calculated true and false positives as well as false negatives by thresholding the IoU. Then the average and standard deviation of the resulting metrics were calculated. As only 13 polygons out of 3674 (0.35%) were labeled as group and they covered only small areas, we treated them as nonexistent during evaluation.

5 Results

5.1 Satellite imagery

In the first scenario on urban trees using satellite imagery, we achieved an accuracy of 46.3%, a precision of 62.8%, and a recall of 63.7% on the test dataset. The IoU of the tree cover mask and its prediction was 71.2%. Considering single trees only, the model found 64.8% of them correctly. In absolute values, the test dataset contained 799 trees and tree groups, of which our method detected 502 correctly, 297 were false positives, and 286 were false negatives. On average, the correctly detected polygons had an intersection over union with their corresponding reference polygon of 71.2%, a precision of 84.6% and a recall of 83.4%. The prediction for the entire study area of 250 km^2 yielded approximately 0.55 million tree and tree group polygons. Inference took 35 minutes on a workstation with two Nvidia GTX1080Ti, 96 GB RAM and a solid-state hard drive. Figure 3 shows three examples: an urban context from the test set and two from outside our training data but with similar structure.

Table 2 Reference datasets used for the two experiments

	Satellite	Aerial
Tree count	9660	3674
Training images	28/35	32
Validation images	7	7
Test images	23	–
Training image size	300 × 300 m	50 × 50 m
Resolution	30 cm	5 cm
Cross-validation	–	10-fold

The reference dataset distinguished single trees and tree groups, which could not be separated visually in the satellite image. Figure 4 depicts the relationship between the size of single tree crowns (or groups) and the recall, answering how many of the in situ measured tree crowns were found correctly. The results show that the method performs poorly for trees with crown areas below 10 m^2 with a recall of only 7.4%. Larger trees, however, were detected with much higher reliability, with a recall of more than 80% for single trees and between 67 and 86% for tree groups. The decreasing recall for large tree groups indicates that the model failed to partition the groups correctly. Note that the uncertainty grows with decreasing tree count for the corresponding size class.

5.2 Aerial imagery

In the second scenario, we analyzed the model performance in aerial images, focusing on single trees only. Tenfold cross-validation was performed, so all following metrics and values were averaged over these runs. We reached a maximum average accuracy of $52.0 \pm 3.8\%$, with an average precision and recall of $70.9 \pm 3.1\%$ and $66.2 \pm 5.3\%$, respectively. The intersection over union of the tree cover mask reached $81.9 \pm 2.2\%$, which indicates that tree / non-tree areas are well separated. Looking at the quality of correctly predicted polygons in comparison to their reference polygon counterpart, the average IoU was $72.6 \pm 1.2\%$, the precision was $90.6 \pm 1.2\%$ and the recall a $79.6 \pm 1.8\%$. Inference on the whole dataset of 142 km^2 took 7.5 hours and resulted in approximately 3.7 million tree crown polygons. Figure 5 depicts the results obtained on the validation set of one cross-validation run and for two scenarios outside of our training and validation sets:

The outputs for medium-sized (20–60 m^2) deciduous trees agreed well with visual perception. The same was observed for most coniferous trees that were visually well separable in the images. Figure 6 shows that the model performance again depended on the tree crown size. Forty percent of the trees smaller than 10 m^2 were detected

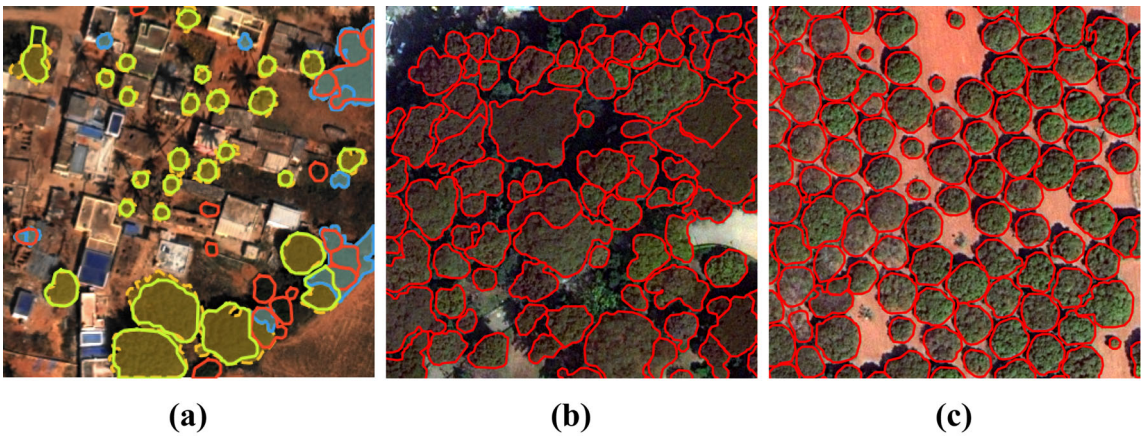

Fig. 3 Results from scenario 1: **a** An urban region from the test set. True positives are green, false positives red, and false negatives blue. Reference data are shown in yellow. **b** Heterogeneous canopy cover and **c** a mango plantation

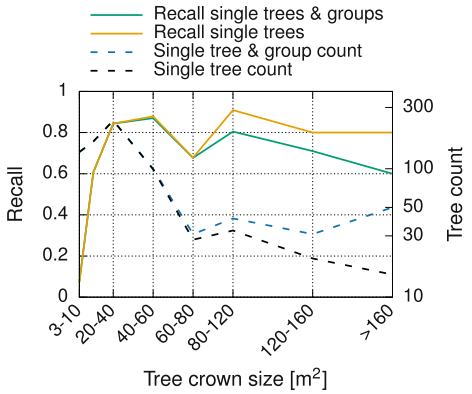

Fig. 4 The recall for different tree and tree group sizes. At the given resolution of 30 cm, the model performed poorly for trees smaller than 10 m^2. For trees between 20 and 160 m^2, the model correctly delineated 67–86% of the tree groups and 67–91% of the single trees. For even larger trees, the recall dropped. Note that the uncertainty increases with decreasing tree count

correctly and the maximum recall of 85% was reached for tree crowns between 20 and 40 m^2 in size. Between 10 and 60 m^2, the recall was higher than 75%.

However, the results show that large tree crowns tend to be over-segmented. To quantify this effect, we calculated the over-segmentation ratio (see Eq. (6)) depending on the crown size. Figure 7 supports our findings and shows that for tree crowns larger than 60 m^2, there are on average 2.4 polygons per tree. By choosing a larger minimum distance of 3 m instead of 2 m, this can be reduced to about 1.6 polygons on average, while increasing the recall from 48 to 68% at the same time. However, simultaneously the accuracy is reduced and smaller trees are no longer detected.

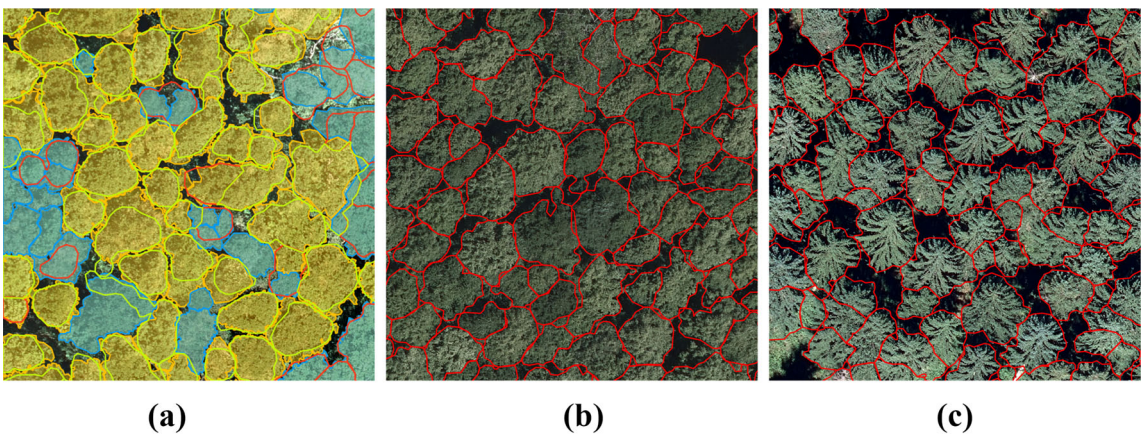

Fig. 5 Method output for three scenarios: **a** Generated tree crown polygons from the validation set of one k-fold run. True positives are green, false positives are red, false negatives are blue, and reference data are yellow. **b** The results in mixed, closed-canopy deciduous forests agreed well with visual perception and **c** spruce were separated well

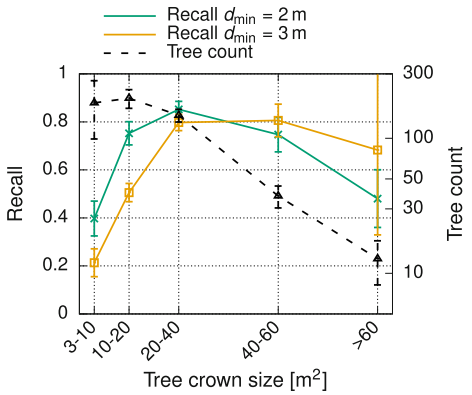

Fig. 6 Recall depending on tree size. At this resolution, our method works best for trees between 10 and 60 m^2

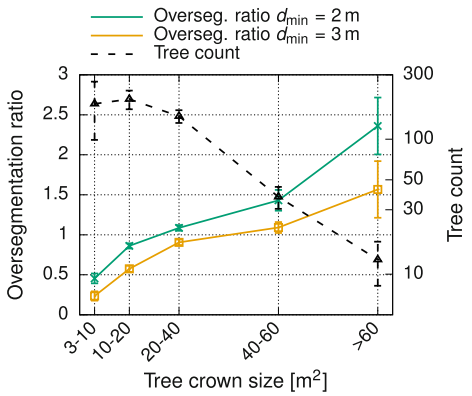

Fig. 7 Large trees tend to be over-segmented, which is remedied by increasing the minimum tree distance—at the cost of decreased overall accuracy

6 Discussion

The presented method delineates tree crowns under different environmental conditions and with different sensors. The average precision (70.9%) and recall (66.2%) reached in our aerial image dataset, are comparable to those obtained by [22], who worked with aerial imagery (10 cm spatial resolution) of various locations. Braga et al. [25] reached an overall detection accuracy of 91% within satellite imagery (50 cm spatial resolution), compared to 46.3% accuracy we reached with our satellite data. However, both references used the intersection over union of tree crown bounding boxes as evaluation criterion. In contrast, we used the IoU of irregularly shaped tree crown polygons, which is a stricter criterion.

We analyzed two case studies and achieved a comparable accuracy with both datasets, although the spatial resolution of the imagery differed (30 cm vs 5 cm) and one might expect higher accuracy from higher resolution. In the following we show that, depending on the dataset, there are different factors affecting accuracy and explain the limitations of our method.

Satellite imagery: In the satellite imagery dataset, three main shortcomings of the model itself and the labels are apparent: (1) Small trees were sometimes missed, (2) the network was not able to separate homogeneous closed canopy cover, and (3) labeling contiguous crowns as tree groups induced ambiguities.

Figure 4 shows that the recall for tree crowns with sizes between 3 and 10 m^2 is below 10% which is confirmed by Figure 8a where many small trees were not detected. Such small trees cover only approximately 10 by 10 pixels and were therefore hard to detect in the satellite images. Figure 8b shows homogeneous, closed canopy cover, which was not correctly delineated. However, this is only partly a deficiency of the model, as delineating such areas is extremely difficult for humans, too. Even with in situ data from ground measurements, it is in some cases impossible to separate individual crowns, due to them growing into each other. At this point, the concept of individual crown delineation stops being applicable. As consequence, we labeled adjacent tree crowns as a group if they were indistinguishable from the ground and on screen. However, this made it difficult to infer the actual accuracy because, for example, the network has not labeled the center tree group in Figure 8c correctly—it has split the group into several tree crowns, which is actually closer to reality than our labels. Therefore the network over-segments from the label-perspective but under-segments the real situation on the ground.

Aerial imagery: In our aerial imagery, ambiguous tree group labels were less of an issue compared to the first case study. Small trees (3–10 m^2) were detected with higher recall, as trees were better separable due to the higher image resolution. Instead, the most prominent issue was the over-segmentation of very large tree crowns. The model recall decreased for large (> 60 m^2) deciduous trees and damaged or leafless deciduous trees (see Figs. 7 & 9). We attribute the over-segmentation to three causes: (1) a lack of training data for large trees, as they are quite rare, (2) the watershed segmentation itself is prone to over-segmentation [34], and (3) the inherent difficulty of the task; segmenting e.g., large, disjunct oak crowns is also hard for humans and often ambiguous, which becomes even harder when the tree crown is damaged. However, our method can be adapted to stands with homogeneously sized trees by adjusting the polygon extraction parameters to mitigate over-segmentation and maximize performance for certain tree crown sizes.

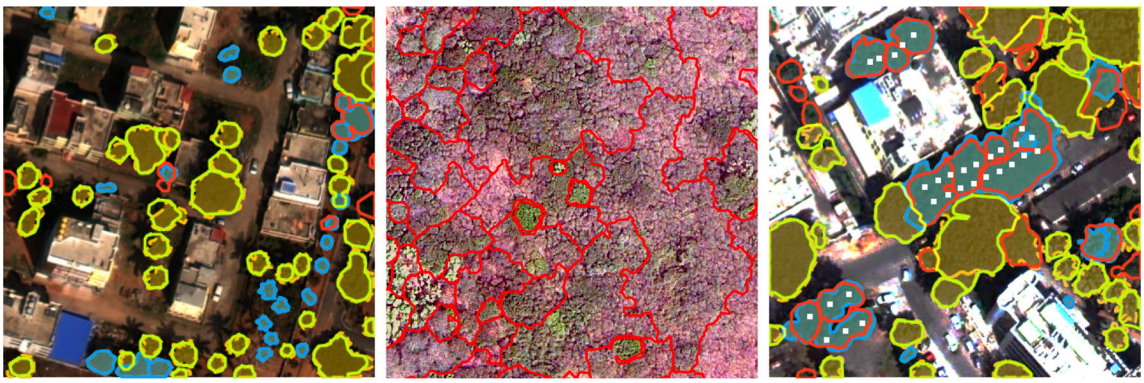

Fig. 8 Three situations where tree crowns were not correctly delineated: **a** very small trees, **b** homogeneous canopy cover with no visible crowns, and **c** comparison of field measured tree stem positions (white boxes) with crown segments. Green lines indicate true positives, blue false negatives, and red false positives. Reference data are shown in yellow

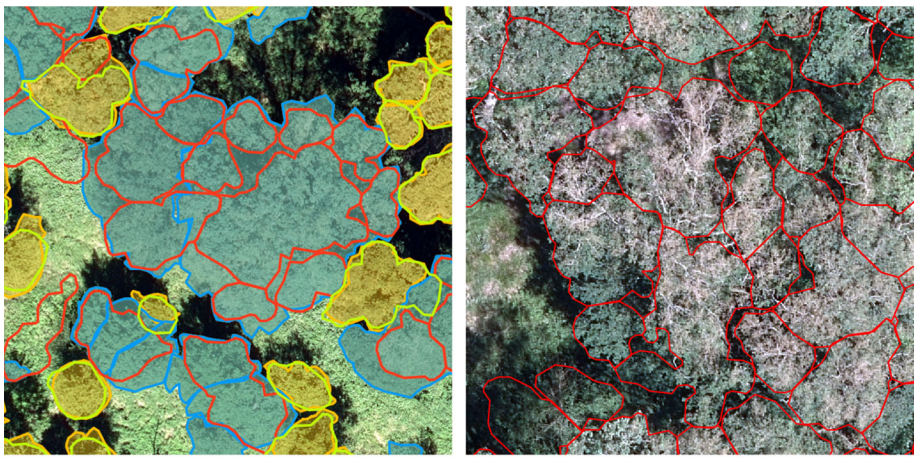

Fig. 9 Examples of over-segmentation of large deciduous trees: **a** large oak crowns, split into many polygons **b** incorrect delineation of damaged trees

7 Conclusion

We presented a method for delineating individual tree crowns or groups of tree crowns without taking 3D data (LiDAR) into account. The method was tested in two scenarios: (1) In WorldView-3 satellite imagery of a 250 km^2 large area around Bengaluru, India. In this case, the imagery had 30 cm spatial resolution and 8 spectral bands ranging from blue to infrared. (2) In aerial imagery of a 142 km^2 large forested area near Gartow, Germany. Here, the imagery had red, green, blue and near infrared color channels and 5 cm spatial resolution. The approach reached satisfying results in both experimental setups and has three major advantages: (1) it requires only a small training dataset, (2) it is fast when predicting, which renders it applicable to entire regions or even countries, and (3) it is applicable to different data sources (satellite/aerial) and environments (urban trees/dense forests). Due to the low training data requirements, it can be a valid option for smaller scale studies with limited access to training data or labeling capacities. As future extension, we plan to implement simultaneous classification and delineation, as well as to improve the neural network performance by pretraining with training data generated from LiDAR canopy height models—the inference will still be able to run on optical imagery alone. Altogether, the developed method can be an important contribution to not only improve detailed forest monitoring (e.g., enabling the detection of selective logging), but also to provide economical value to forest owners who could use this tool for better yield estimation.

Appendix 1 Metrics

These equations describe the metrics used for evaluating the model performance:

$$\text{accuracy}\,(\text{tp},\text{fp},\text{fn}) = \frac{\text{tp}}{\text{tp}+\text{fp}+\text{fn}} \tag{7}$$

$$\text{precision}\,(\text{tp},\text{fp}) = \frac{\text{tp}}{\text{tp}+\text{fp}} \tag{8}$$

$$\text{recall}\,(\text{tp},\text{fn}) = \frac{\text{tp}}{\text{tp}+\text{fn}} \tag{9}$$

tp: number of true positive polygons, fp: false positives, fn: false negatives

Acknowledgements We thank the family von Bernstorff for supporting this project and providing the aerial images of Gartow, Germany.

Author Contributions Maximilian Freudenberg was responsible for code development and for the first draft of the manuscript. All authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL. This research was in part funded by the German Research Foundation, DFG, through grant number KL 894/23-2 and NO 1444/1-2 and by the German Federal Ministry for Digital and Transport under grant number 50EW2012B.

Data availibility The source code and pretrained weights are publicly available at https://github.com/AWF-GAUG/TreeCrownDelineation. The data used in this work is subject to restrictive licensing terms, which do not allow sharing. An anonymized data sample is in preparation and will be published along with the code.

Declarations

Conflicts of interest The authors declare that they have no conflicts of interest.

References

1. Dalponte M, Frizzera L, Ørka HO, Gobakken T, Næsset E, Gianelle D (2018) Predicting stem diameters and aboveground biomass of individual trees using remote sensing data. Ecol Indic 85:367–376. https://doi.org/10.1016/j.ecolind.2017.10.066
2. Wyckoff PH, Clark JS (2005) Tree growth prediction using size and exposed crown area. Can J For Res 35(1):13–20. https://doi.org/10.1139/x04-142
3. Pommerening A, Gaulton R, Magdon P, Myllymäki M (2021) Canopyshotnoise-an individual-based tree canopy modelling framework for projecting remote-sensing data and ecological sensitivity analysis. Int J Remote Sens 42(18):6837–6865. https://doi.org/10.1080/01431161.2021.1944695
4. Getzin S, Wiegand K, Schöning I (2012) Assessing biodiversity in forests using very high-resolution images and unmanned aerial vehicles. Methods Ecol Evol 3(2):397–404. https://doi.org/10.1111/j.2041-210X.2011.00158.x
5. Lamar WR, McGraw JB, Warner TA (2005) Multitemporal censusing of a population of eastern hemlock (tsuga canadensis l.) from remotely sensed imagery using an automated segmentation and reconciliation procedure. Remote Sensing of Environment 94(1):133–143. https://doi.org/10.1016/j.rse.2004.09.003
6. Brandtberg T (1999) Automatic individual tree based analysis of high spatial resolution aerial images on naturally regenerated boreal forests. Can J For Res 29(10):1464–1478. https://doi.org/10.1139/x99-150
7. Skurikhin AN, Garrity SR, McDowell NG, Cai DM (2013) Automated tree crown detection and size estimation using multi-scale analysis of high-resolution satellite imagery. Remote Sens Lett 4(5):465–474. https://doi.org/10.1080/2150704X.2012.749361
8. Erikson M (2003) Segmentation of individual tree crowns in colour aerial photographs using region growing supported by fuzzy rules. Can J For Res. https://doi.org/10.1139/x03-062
9. Ke Y, Quackenbush LJ (2011) A review of methods for automatic individual tree-crown detection and delineation from passive remote sensing. Int J Remote Sens 32(17):4725–4747. https://doi.org/10.1080/01431161.2010.494184
10. Lindberg E, Holmgren J (2017) Individual tree crown methods for 3d data from remote sensing. Curr For Rep 3(1):19–31. https://doi.org/10.1007/s40725-017-0051-6
11. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969. 10.1109/ICCV.2017.322
12. Iglovikov VI, Seferbekov S, Buslaev AV, Shvets A (2018) TernausNetV2: fully convolutional network for instance segmentation. arXiv:1806.00844 [cs] 1806.00844
13. Li Y, Zhao H, Qi X, Wang L, Li Z, Sun J, Jia J (2021) Fully convolutional networks for panoptic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 214–223
14. Pereira GHdA, Fusioka AM, Nassu BT, Minetto R (2021) Active fire detection in landsat-8 imagery: a large-scale dataset and a deep-learning study. arXiv preprint arXiv:2101.03409
15. Shi Q, Liu M, Li S, Liu X, Wang F, Zhang L (2021) A deeply supervised attention metric-based network and an open aerial image dataset for remote sensing change detection. IEEE Trans Geosci Remote Sens 60:1
16. Zhang P, Ke Y, Zhang Z, Wang M, Li P, Zhang S (2018) Urban land use and land cover classification using novel deep learning models based on high spatial resolution satellite imagery. Sensors 18(11):3717
17. Xi Y, Ren C, Tian Q, Ren Y, Dong X, Zhang Z (2021) Exploitation of time series sentinel-2 data and different machine learning algorithms for detailed tree species classification. IEEE J Sel Top Appl Earth Obs Remote Sens 14:7589–7603

18. Zhang C, Xia K, Feng H, Yang Y, Du X (2021) Tree species classification using deep learning and rgb optical images obtained by an unmanned aerial vehicle. J For Res 32(5):1879–1888
19. Hartling S, Sagan V, Sidike P, Maimaitijiang M, Carron J (2019) Urban tree species classification using a worldview-2/3 and lidar data fusion approach and deep learning. Sensors 19(6):1284
20. Plesoianu A-I, Stupariu M-S, Sandric I, Pătru-Stupariu I, Drăgut L (2020) Individual tree-crown detection and species classification in very high-resolution remote sensing imagery using a deep learning ensemble model. Remote Sens 12(15):2426. https://doi.org/10.3390/rs12152426
21. Hao Z, Lin L, Post CJ, Mikhailova EA, Li M, Chen Y, Yu K, Liu J (2021) Automated tree-crown and height detection in a young forest plantation using mask region-based convolutional neural network (mask r-cnn). ISPRS J Photogramm Remote Sens 178:112–123. https://doi.org/10.1016/j.isprsjprs.2021.06.003
22. Weinstein BG, Marconi S, Aubry-Kientz M, Vincent G, Senyondo H, White EP (2020) DeepForest: a python package for RGB deep learning tree crown delineation. Methods Ecol Evol 11(12):1743–1751. https://doi.org/10.1111/2041-210X.13472
23. Weinstein BG, Marconi S, Bohlman S, Zare A, White E (2019) Individual tree-crown detection in rgb imagery using semi-supervised deep learning neural networks. Remote Sens 11(11):1309. https://doi.org/10.3390/rs11111309
24. Lin T-Y, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp 2980–2988. https://doi.org/10.1109/TPAMI.2018.2858826
25. Braga GJR, Peripato V, Dalagnol R, Ferreira PM, Tarabalka Y, Aragão OCLE, de Campos Velho FH, Shiguemori EH, Wagner FH (2020) Tree crown delineation algorithm based on a convolutional neural network. MDPI 12(8):1288. https://doi.org/10.3390/rs12081288
26. Bai M, Urtasun R (2017) Deep watershed transform for instance segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 5221–5229
27. Ronneberger O, Fischer P, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-assisted Intervention, pp 234–241. https://doi.org/10.1007/978-3-319-24574-4_28
28. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 770–778
29. Yakubovskiy P (2020) Segmentation Models Pytorch. GitHub. https://github.com/qubvel/segmentation_models.pytorch
30. Bergstra J, Yamins D, Cox D (2013) Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In: International Conference on Machine Learning, pp 115–123. PMLR
31. Van der Walt S, Schönberger JL, Nunez-Iglesias J, Boulogne F, Warner JD, Yager N, Gouillart E, Yu T (2014) scikit-image: image processing in python. PeerJ 2:453
32. Loshchilov I, Hutter F (2016) Sgdr: Stochastic gradient descent with warm restarts. arXiv:1608.03983 1608.03983
33. Brandt M, Tucker CJ, Kariryaa A, Rasmussen K, Abel C, Small J, Chave J, Rasmussen LV, Hiernaux P, Diouf AA et al (2020) An unexpectedly large count of trees in the west african sahara and sahel. Nature 587(7832):78–82. https://doi.org/10.1038/s41586-020-2824-5
34. Beucher S (1994) Watershed, hierarchical segmentation and waterfall algorithm. Math Morphol Appl Image Process. https://doi.org/10.1007/978-94-011-1040-2_10