Accurate tropical forest individual tree crown delineation from aerial RGB imagery using Mask R-CNN

James G. C. Ball[*a,b,c], Sebastian H. M. Hickman[*d,e], Tobias D. Jackson[*a,b], Xian Jing Koay[b], James Hirst[f], William Jay[g], Mélaine Aubry-Kientz[h], Grégoire Vincent[c], and David A. Coomes[a,b] ✉

[a]Department of Plant Sciences, University of Cambridge, Downing Street, Cambridge, CB2 3EA. United Kingdom.
[b]Conservation Research Institute, University of Cambridge, Downing Street, Cambridge, CB2 3EA. United Kingdom.
[c]UMR AMAP, University of Montpellier, IRD, CNRS, CIRAD, INRAE, Montpellier, France
[d]Yusuf Hamied Department of Chemistry, University of Cambridge, Lensfield Road, Cambridge, CB2 1EW. United Kingdom.
[e]The Alan Turing Institute, 96 Euston Road, London, NW1 2DB. United Kingdom.
[f]Department of Applied Mathematics and Theoretical Physics, Wilberforce Road, Cambridge, CB3 0WA, United Kingdom
[g]Plymouth Marine Laboratory, Prospect Place, Plymouth, PL1 3DH. United Kingdom.
[h]AgroParisTech, UMR EcoFoG, Kourou, French Guiana
[*]**Authors contributed equally to the work (first authors)**

1. Tropical forests are a major component of the global carbon cycle and home to two-thirds of terrestrial species. Upper-canopy trees store the majority of forest carbon and can be particularly vulnerable to drought events and storms. Monitoring their growth and mortality is essential to understanding forest resilience to climate change, but large trees are underrepresented in traditional field surveys, so estimates are poorly constrained.

2. Aerial photographs provide spectral and textural information to discriminate between tree crowns in diverse, complex tropical canopies, potentially opening the door to landscape monitoring of large trees. Here we describe a new deep convolutional neural network method, *Detectree2*, which builds on the Mask R-CNN computer vision framework to recognise the irregular edges of individual tree crowns from airborne RGB imagery. We trained and evaluated this model with 3,800 manually delineated tree crowns at three sites in Malaysian Borneo and one site in French Guiana. As an example application of this tool, we combined the delineations with repeat lidar surveys of the four sites to estimate the growth and mortality of upper-canopy trees.

3. *Detectree2* delineated 65,000 upper-canopy trees across 14 km^2 of aerial images. The skill of the automatic method in delineating unseen test trees was good (F_1 score = 0.64) and for the tallest category of trees was excellent (F_1 score = 0.74). As predicted from previous field studies, we found that growth rate declined with tree height and tall trees had higher mortality rates than intermediate-size trees. In addition, trees in French Guiana had higher growth and mortality rates than those in Borneo.

4. Our approach demonstrates that deep learning methods can automatically segment trees in widely accessible RGB imagery. This tool (provided as an open source Python package) has many potential applications in forest ecology and conservation, from estimating carbon stocks to monitoring forest phenology and restoration. We demonstrate its use in tracking the growth and mortality rates of upper-canopy trees at scales much larger than those achievable with field data.

Python package available to install at `https://github.com/PatBall1/Detectree2`

Tropical forests | Tree crown delineation | Tree crown segmentation | Deep learning | Convolutional neural networks | Mask R-CNN | Detectron2 | Tree growth | Tree mortality | Forest monitoring

Correspondence: ***ball.jgc@gmail.com; shmh4@cam.ac.uk;***

Introduction

Intact tropical forests are an important component of the global carbon cycle: they are major carbon stores and significant carbon sinks (1). However, the strength of the carbon sink is diminishing as a result of global warming (2, 3) and there are concerns that forests are reaching a tipping point beyond which they could switch irreversibly to open savanna systems (4). However, forecasting the future of tropical forests is challenging, because little is known about the ways different species will respond to changing climate, or the resilience provided by that diversity (5–8). To understand the likely responses of forests to further climate change, ecosystem models need to represent growth and mortality processes of individual trees more accurately than is currently the case (9–11).

Remote sensing of individual upper-canopy trees can improve estimates of forest carbon (12) and provide a means of tracking growth and mortality over large spatial scales. Traditional monitoring approaches rely on measuring stem dimensions in permanent inventory plots, and periodically revisiting those plots to assess recruitment, growth and mortality (13). However, the coverage of such plots in the tropics is limited (~0.0002% of tropical forests are sampled by the main plot networks) and their locations are often dictated by ease of access rather than by robust statistical sampling designs (14–16). Furthermore, upper-canopy trees store the majority of carbon in tropical forests, but few of them are sampled in inventory plots (17–19). This under-sampling is particularly problematic when assessing impacts of climate change, because upper-canopy trees are most vulnerable to periods of water shortage (20, 21) which are increasing in frequency (22). Remote sensing has the potential to overcome these sampling challenges by providing wall-to-wall maps that can be used to monitor millions of upper-canopy trees.

Remote sensing of individual trees has focused on airborne lidar data, which is used by the forestry industry to map trees at landscape scales (23). Delineating individual trees from airborne lidar datasets is most successful for coniferous forests, because their apical dominance results in clear local height maxima that make tree crowns easily distinguishable

DRAFT

(12, 24), but complex tropical canopies have presented a far greater challenge for lidar delineation (25). Tropical forest canopies are often densely packed with partially interwoven crowns which point-cloud clustering algorithms can struggle to distinguish (26). Furthermore, lidar surveys require expensive aircraft (airplanes, helicopters or high-end drones) and sensors whereas standard RGB imagery can be collected cheaply with low-end drones.

Automatic delineation of trees in RGB photographs can harness colour and texture information to distinguish trees, even if they are structurally similar (27, 28). Most current methods of individual tree identification from RGB imagery use bounding boxes (Fig. S3) (29–31), but more exact delineation of the edges of tree crowns would provide information on crown area and lateral growth, and avoid mixing of signals from neighbouring trees. Recent advances in neural network approaches to computer vision provide opportunities to recognise individual trees from standard digital photographs taken from drones. A class of models machine learning algorithms referred to as deep convolutional neural networks (CNNs) is revolutionising vegetation science (32) through its ability to exploit spatial structures and automatically extract high-level features from image data (e.g. analyses of satellite imagery (33)). In the field of computer vision, exactly segmenting individual objects of interest from an image is known as *instance segmentation*. The Mask R-CNN algorithm has shown promise in tree crown identification and delineation in plantations (34, 35), pine forests (36, 37), urban woodlands (36) and forest fragments (38). Here we use Facebook AI's Mask R-CNN algorithm, (the *Detectron2* release) (39, 40), which has models that have been pre-trained on a wealth of available image data that can be transferred to new tasks. The Detectron2 framework has features that could allow it to overcome the challenges of delineating crowns in dense, complex tropical canopies by discriminating based on the spectral and textural signals which are rich due to the phylogenetic diversity.

In this paper, we describe Detectree2, an adaptation of Facebook AI's Detectron2 system that detects tree crowns from aerial RGB imagery. We trained and evaluated Detectree2 on four tropical forest sites. In total, 3797 manually delineated tree crowns were used of which 1530 spatially separated crowns were reserved to evaluate the model. We evaluated the performance with F_1 scores, which quantify the skill of the method in delineating individual tree crowns accurately. We expected a model trained at one site to drop in performance when transferred to making predictions of crowns at the other sites. We tested the effect of including different combinations of training sites on prediction accuracy accuracy expecting that a greater variety of training data would boost performance. As an example of its ecological application, we deployed the trained model across 14 km^2 of airborne RGB imagery, automatically delineating 65,786 tree crowns. For context, this area is approximately 40% the total area of tropical forest inventory plots in the main plot networks (14, 15) We then combined these predicted crowns with repeat airborne lidar data to investigate the growth and mortality rates of upper-canopy trees. Regional and global syntheses of forest inventory data suggest that growth slows down and mortality rates increase with tree size (41–45). We therefore expected to find the tallest trees we sampled to have lower growth rates and higher mortality rates than shorter trees. To explore this, we calculated the change in height for every tree and classed large negative changes as mortality events. We then tested whether growth or mortality events varied with initial tree height. The Detectree2 Python package is available to install and apply on new regions[1].

Materials and Methods

Study sites. The analyses were conducted at four locations across three tropical field sites:

1. Sepilok Forest Reserve (East and West), Sabah, Malaysia (5° 50' N, 177° 56' W)
2. Danum Valley Conservation Area, Sabah, Malaysia (4° 57' N, 177° 41' W)
3. Paracou Field Station, French Guiana (5° 16' N 52 ° 55' W)

Danum Valley hosts lowland tropical rain forests dominated by dipterocarp species that are among the tallest forests on the planet (46). The available data from Sepilok included ecologically distinctive areas to the East and West. Sepilok West consists mostly of tall forest (similar to Danum), while Sepilok East is a heath forest growing on shallow soils overlying sandstone, containing smaller, more densely packed trees (19). All three sites in Malaysia experience a similar climate with approximately 2300 mm rainfall per year with the wettest months between November and February (47). Paracou contains lowland tropical rain forest mostly on shallow ferralitic soils that lay on a variably transformed loamy saprolite (48). The mean annual rainfall is approximately 3000 mm with a three month dry season from mid-August to mid-November (49). See Sup. Note 1 for more details on the study sites.

Remote sensing data. Airborne RGB surveys were conducted at all four sites using manned aircraft (Table 1). Repeat lidar surveys were also conducted at all four location (see Table 1, noting different sensors and altitudes between flights in Sabah). We analysed RGB imagery from 3.85 km^2 of Malaysian forest, with a ground resolution of 10 cm. In Paracou, we sampled 10.2 km^2 of imagery, with an 8 cm ground resolution. The raw imagery was orthorectified and collated into homogeneous mosaics using AgiSoft Metashape (50). This software uses structure from motion to generate a 3D elevation surface from the raw aerial photographs (51) which is then projected back into 2D to create a landscape scale image (orthomosaic). This gave georeferenced photographs, largely corrected for distortions. Some distortion remained at the edges of the mosaic, so the analysis focused on the core areas where the imagery was not distorted.

[1] https://github.com/PatBall1/Detectree2

DRAFT

Table 1. Remote sensing data sources. Resolution is given as ground resolution for the RGB imagery and returns per m^2 for lidar scans. The Sabah region scans include both Sepilok and Danum sites.

Date	Region	Modality	Res.	Altitude	Sensor
10/2014	Sabah	RGB	10 cm	796 m	Leica RCD105
10/2014	Sabah	lidar	7.3 m^{-2}	2000 m	Leica ALS50
10/2020	Sabah	lidar	16 m^{-2}	250 m	Reigl LMS-Q780
19/09/2016	Paracou	RGB	8 cm	800 m	IXA180 Phase One
19/09/2016	Paracou	lidar	50 m^{-2}	800 m	RIEGL LMS-Q780
15/11/2019	Paracou	lidar	50 m^{-2}	800 m	RIEGL LMS-Q780

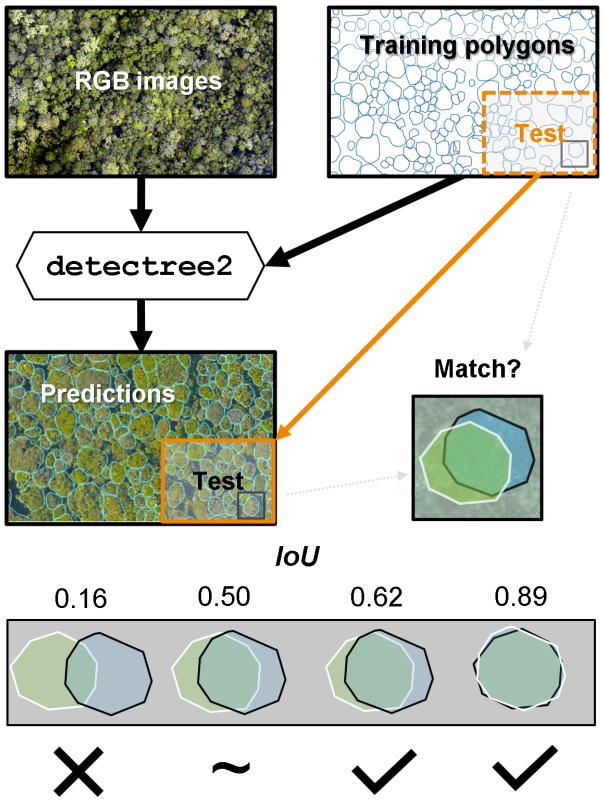

Fig. 1. The automatic tree crown delineation workflow. Manually delineated crowns are split into training and test sets. The Mask R-CNN framework combines the training set with the RGB imagery to learn how to delineate automatically from RGB images. A set of automatic predictions are produced across the entire RGB image and compared to the test set to evaluate the performance of the automatic delineations. IoU is used to determine when an automatic crown has been successfully matched with a manual crown.

Manual tree crown data. To train and assess the accuracy of our automatic delineation approach, we created a manually labelled dataset of trees at all four sites. We generated our delineations using both RGB and lidar data and, in the case of Paracou, supplementary hyperspectral layers. We used a number of techniques to improve the accuracy of crown delineation, including altering the contrast and saturation of the RGB image to exaggerate differences between the crowns, and using a mask of the lidar data to remove irrelevant parts of the RGB imagery. See Sup. Note 2 for further details.

We trained and tested our model with a total of 3797 manually delineated tree crowns across Paracou (1267), Danum (521), Sepilok West (1038) and Sepilok East (971). The crowns from Paracou were validated in the field with an expert local botanist, whereas the crowns in Malaysia were drawn by inspection of the remote sensing products. Four individuals performed the manual delineations which provided the network with variability in the inputs. This variability can be viewed as a form of data regularisation. Models trained on this variety could therefore be seen as a synthesis of the human delineators, limited by their skill but harmonising their perspectives into a consistent output.

Data preparation. The orthomosaics and corresponding crown polygons were tiled into squares of approximately 100 m x 100 m to be ingested into the network (40 m core area, 30 m overlapping buffers). To be included in the training and test sets, a minimum crown polygon area coverage of a tile was set at 40%. Including overly sparse tiles was likely to lead to poor algorithm sensitivity while being too strict with coverage would have limited the amount of training and testing data available.

If training and test crowns are close to one another, spatial autocorrelative effects are likely to inflate the reported performance (52). To avoid this, individual tiles (rather than individual crowns) were assigned to training and test sets ensuring spatial separation. Approximately 10% of the tiles from each site were reserved at random for testing. To avoid contamination of the test set, tiles with any overlap with the test tiles (including with the buffer) were excluded from the training set. The training tiles were further partitioned into 5-folds for cross validation. This allowed for the tuning of parameters and the implementation of early stopping (see **Training and model selection**) without exposing the test set. Details of the data processing are described in Sup. Note 3.

Model architecture and parameterisation. Instance segmentation combines object detection with object segmentation. Once an object has been detected in a scene, a region of interest (as a bounding box) is established around the object. Then a "segmentation" is then carried out to identify which pixels within the region of interest make up the object of interest (and which lie outside; see Fig. S2 for an example).

We chose to use Facebook AI's Mask R-CNN architecture as it was the best in class algorithm upon release for instance segmentation when tested on the Microsoft COCO (Common Objects in Context) benchmark (39, 53) and has since been updated (as Detectron2) with improved training efficiency, documentation and transferability for integration into bespoke tools (40). We adapted the Detectron2 computer vision library to facilitate the delineation of individual tree

DRAFT

crowns. The library performs instance segmentation by identifying and locating objects within images while also generating object "masks" which exactly circumscribe the objects in the image (see Fig. S2 for an example prediction). It also has a "model zoo"[2] from which specific model architectures with weights from a variety of pre-training regimes can be loaded. Taking a pre-trained model (weights) and retraining it to perform a novel task (e.g. delineating trees from aerial imagery) is an example of *transfer learning* which can drastically reduce the amount of training data required to achieve acceptable performance on the new task (54).

We selected the R101-FPN configuration[3] as it has "the best speed/accuracy tradeoff" of the architectures available (40). The configuration sets the *backbone* of the network which is the part that views and extracts features from the scene as a whole. The R101-FPN backbone consists of a 101 layer deep ResNet (55) module with a Feature Pyramid Network (56) module. The initial model weights were generated from pre-training of the network on the ImageNet dataset[4]. It is possible to "freeze" the backbone to different depths depending on the amount of flexibility the user wants to introduce in moving away from the pre-trained model weights. Each object predicted by Detectron2 is associated with a confidence score which relates to how sure the network is in its prediction. A suitable threshold can be selected to optimise accuracy or balance precision and recall. Additional details are given in Sup. Note 4 and for full technical specifications, one should refer to the original papers and the Detectron2 repository (39, 40).

Training and model selection. Training, tuning and model selection were performed with the five folds of training data tiles (see **Data preparation**). To test the effect of volume and diversity of training data on performance we employed three training regimes: (1) Training on data of a single site, (2) Training on all sites, (3) Training on all sites and then trained with a fixed training period on the single site. The idea behind (3) was to train on the full available data and then 'hone' the delineator based on the local context.

Typically, a deep CNN would require several thousand training examples to learn a new task. This is a challenge in the case of tree crown delineation as manual delineation is very time consuming. The burden was reduced by transferring a model trained on a different instance segmentation task (53). Additionally, the training data were augmented by applying several randomly applied transformations to the training cases including vertical and horizontal flips, rotation, and varying the saturation and contrast of the image.

The model hyperparameters (see Table S1) were tuned with a Bayesian hyperparameter sweep implemented on wandb.ai[5]. This is an automated process that allows an automated agent to iteratively adjust hyperparameters to optimise based on a given accuracy metric. We selected the *AP50* of the segmentation predictions on the (randomly selected) validation fold as our metric for optimisation. AP50 is the average precision of predictions when a correct match is granted for $IoU > 0.5$.

AP50 was also used as the metric for *early stopping* whereby if the model performance failed to improve for a set number of training iterations (the "patience"), training would be terminated and the best model up to that point would be saved. This is a technique for regularisation and prevents wasted computation. The best performing models and optimal confidence threshold for a given model (see **Model architecture and parameterisation**) were selected based on the F_1 score (see **Performance evaluation**) on the validation fold.

See Sup. Note 4 for more details on model architecture, training and validation. The Colab (Jupyter) notebooks in the GitHub repository[6] illustrate the best practices for training and selecting models.

Transferability across sites. to determine whether Detectree2 would produce models that were able to generalise across different areas of tropical forest of Detectree2, we carried out some tests to evaluate the performance of the model when trained on one site and transferred to others. We compared these performances against the effect of using the "combined" training regimes described in **Training and model selection**.

Performance evaluation. After training and tuning, the best performing models were taken forward to be evaluated against the test set of tiles. Matches between predictions and manual crowns (i.e. true positives) were identified by assessing the degree of spatial overlap between possible pairs. A minimum area threshold for valid crowns was set to 16 m^2. This removed less than 2 % of manual crowns and introduced a level of consistency between sites and between the effort given by the manual delineators. The threshold was small enough to allow for an inclusive analysis of the variation in performance by tree height. Crown overlap was calculated as the area intersection over union (see Fig. 1):

$$IoU(A,B) = \frac{A \cap B}{A \cup B} \quad \textbf{(1)}$$

where A is the automatically delineated crown area and B is the manually delineated crown area. An IoU of an overlapping pair of more than 0.5 was considered a match. This is a commonly used threshold in similar studies (e.g. (25, 35)) that allows for small discrepancies in alignment and outline. These "true positives" as well as the unmatched predictions (false positives) and unmatched manual crowns (false negatives) were used to calculate the precision, recall and F_1 score of the automatic predictions. In rare cases where more than one automatic delineation had an IoU of greater than 0.5 with a manual crown, then the automatic delineation with

[2] https://github.com/facebookresearch/detectron2/blob/main/MODEL_ZOO.md
[3] https://github.com/facebookresearch/detectron2/blob/main/configs/COCO-Detection/faster_rcnn_R_101_FPN_3x.yaml
[4] https://www.image-net.org/
[5] https://wandb.ai/detectree/tune/sweeps/
[6] https://github.com/PatBall1/Detectree2

DRAFT

the greatest IoU was considered a true positive and the others were classed as false positives.

Despite the best efforts of the manual delineators and selecting for tiles with high manual crown coverage, the manual crowns were inevitably an incomplete representation so recall (fraction of relevant instances retrieved) was an insightful metric. However, to ensure balance with precision we used the balanced F_1 score metric to assess and compare the accuracy of the models. This approach is not biased by tree crown area and is widely used in tree crown segmentation studies (35, 38). See Sup. Note 5 for more details on the evaluation metrics.

To evaluate the performance of Detectree2 across tree heights, we assigned a height to each test crown (based on the median value of the initial CHM within the crown) and arranged them into 5 m height bins. The shortest bin (0-5 m) at each site was iteratively merged with the next shortest bins until more than 10 individuals were represented (e.g. at Paracou 17 trees with a median height of 22.97 m fell into the lowest bin of 0-25 m). An equivalent process was used to define the highest bin at each site. The median tree height was calculated within each bin.

Application to monitoring growth and mortality. We applied our best models for each site to their entire tiled orthomosaics (excluding the very edges where distortion is prominent) to generate site wide crown maps. We combined these crown delineations with repeat lidar surveys to determine the height changes in individual trees in our four sites. We determined the relationship between tree height and tree growth by fitting a robust least squares regression (57) to the data. Robust least squares was chosen to minimise the effects of outliers and mortality events on the regression. We note that here we are measuring the vertical growth of trees, instead of the growth in diameter at breast height (DBH) which is traditionally measured in forest inventory data.

To estimate mortality rates, we needed a suitable metric to identify mortality events. We took a statistical approach defining a mortality event as a negative change in height of more than three standard deviations below the robust least squares fit. This allowed for the possibility that a mortality event may uncover another layer of vegetation rather than the forest floor. This choice was ratified by manual inspection of trees meeting this threshold, and confirming that they constituted mortality events.

We note that differences in lidar scanning parameters (pulse density, scanning angle etc.) can bias height estimates (58). Since this analysis is intended as a demonstration of the potential of Detectree2 in ecology and conservation we did not focus on this potential bias in the lidar data. This potential bias is more likely to affect our results on tree height growth (which are on the order of 0.1-0.5 m) rather than tree mortality (which are at least an order of magnitude larger).

Computation. Training deep CNN models can be computationally expensive and benefits from the availability of GPUs. Model training and evaluation was performed on the Google Colab (Pro) platform which employed Intel(R) Xeon(R) CPU at 2.30GHz with 12.8 GB RAM and Tesla P100-PCIE-16GB GPUs. On this platform, model training always completed within 2 hours.

Table 2. Precision, recall and F_1 score of Detectree2 tree crown delineations by site as measured against the manual crowns of the test set tiles. The unweighted means of the metrics across individual sites are given as a summary overall performance.

	# test trees	Precision	Recall	F_1 score
Paracou	381	0.595	0.543	0.568
Danum	278	0.713	0.662	0.687
Sepilok East	167	0.612	0.653	0.632
Sepilok West	704	0.640	0.656	0.648
Average (sum)	*(1530)*	*0.640*	*0.629*	*0.634*

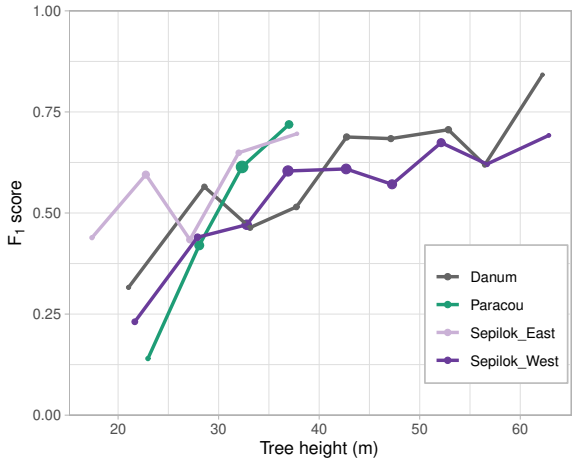

Fig. 2. F_1 scores of the tree crown delineations at the four different sites across tree heights. Bins of 5 m width were used to calculate F_1 score and corresponding median tree height. Point area is scaled by the number of test trees in the bin.

Results

Performance by site and tree height. Detectree2 located and delineated trees well ($F_1 > 0.56$) across all sites (see Table 2). It performed better in the tall dipterocarp dominated forests of Danum and Sepilok West and worse in the more compact forests of Sepilok East and Paracou. Indeed, Danum, the site with the best performance, had the greatest proportion of the tallest class of trees of any of the sites (see **Extended Results** for a full table of results). There was no apparent relationship between the amount of training data available at a site and the performance of the automatic delineator suggesting forest structure was the key determinant of accuracy.

Across all sites, the accuracy of the delineations improved with increasing tree height (see Fig. 2). This is likely due to the increased size and visibility of tall trees in the RGB images, facilitating more straight forward delineation. Interestingly, the accuracy for trees of 30m to 40m was highest at Paracou and Sepilok East, highlighting that differentiated tree height rather than tree height alone is important for accurate delineations. Paracou has the least well differentiated canopy of all sites which may explain why performance was relatively poor there.

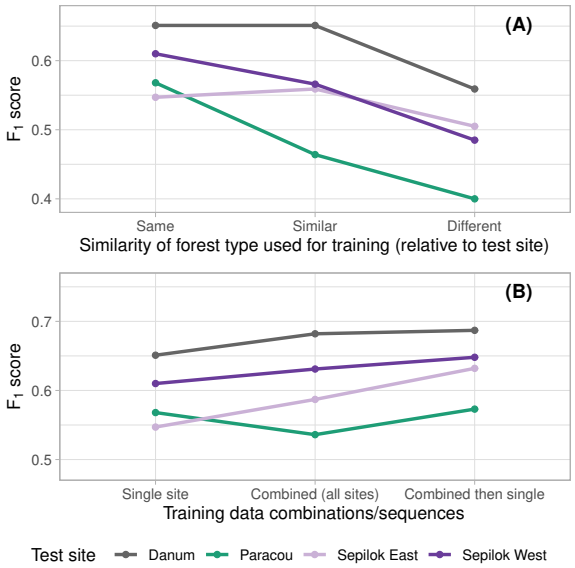

Fig. 3. Sensitivity of Detectree2 delineation accuracy to the training data used. In **(A)**, "Same" indicates that training and testing took place at the same site. Sepilok West and Danum are "similar" forest types in that they are tall dipterocarp dominated forests in contrast to Sepilok East and Paracou that are shorter forests with a larger number of trees per hectare. As each site has two "different" sites and an average was calculated for the F_1 score. **(B)** shows the change in performance that occurs through employing different combinations of the training data. That can be just a single site, all sites at once ("combined") or all sites at once followed by a limited number of iterations on the site to be tested on.

Performance between forest types. As we expected, performance degrades when testing a model on a different forest type to the one it was trained on (see Fig. 3a). For example, the performance at the tall dipterocarp dominated forests of Sepilok West is significantly degraded when a model trained on Sepilok East or Paracou is used. However, when a similar forest type is used in training the performance is much more stable. There is no drop in performance for predictions at Danum when the Sepilok West model is used and there is even a slight increase in performance for Sepilok East predictions when the Paracou model is used.

In general, the model that was trained on all sites at once ("combined") outperformed the models that were trained on just a single site with the exception of Paracou (see Fig. 3b). Across the board, the best performing models were those that were exposed to data from all sites before being trained for a fixed number of iterations at the site to be predicted on. This suggests that providing a broad range of input data helps the networks to learn the key features but further tuning for local context helps maximise performance.

Application: Growth and mortality. One application of Detectree2 is to study tall tree growth and mortality rates. To do this, we overlaid Detectree2's tree crown predictions at the start date for each site on repeat lidar data (as canopy height models described in Sup. Note 1) to retrieve the tree height dynamics over time.

We were able to estimate the relationships between tree height and tree growth for each site by fitting robust least squares linear relationships between the two variables for Danum, Paracou, Sepilok East and Sepilok West. The regression coefficients and intercepts are given in Table S2. We found a higher growth rate in Paracou compared to the three sites in Malaysia. The growth rate decreased with tree height in all sites.

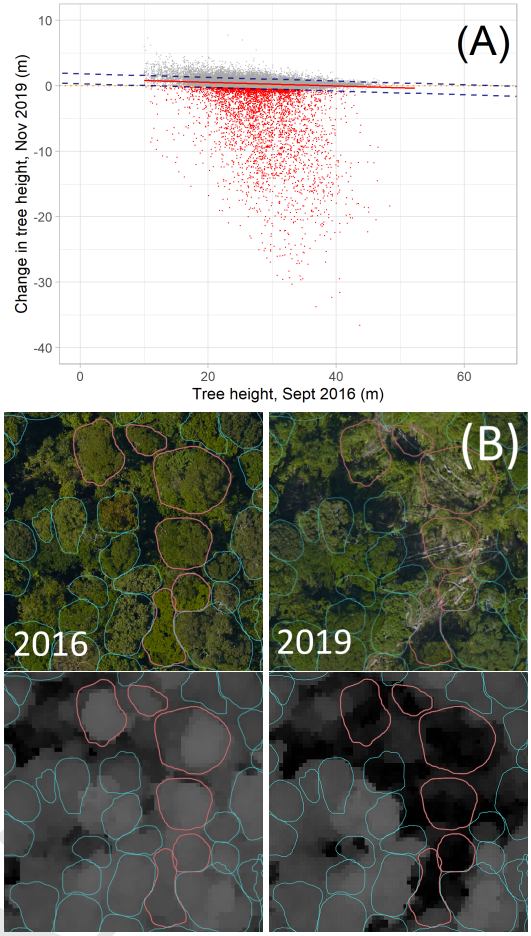

Fig. 4. Plot A shows the robust least squares fit and standard deviations. Plot B illustrates how predictions were overlaid on lidar data, and shows significant mortality events are outlined in red, and shown in the lidar and the RGB imagery. The 2016 imagery is shown on the left, 2019 on the right. Note that the crown delineations are based on the earlier imagery.

We assumed trees had died when their height decreased substantially. The threshold for the required height decrease was calculated by fitting a robust least squares regression to the height change against the original height of the tree, and taking trees that were 3 standard deviations below the mean of the regression fit to be mortality events. Fig. 4 illustrates how certain trees were identified as mortality events and some visual examples of mortality events from Paracou are given alongside.

Using this threshold for mortality events, we found overall annual mortality rates of 1.3% in both Sepilok West and Danum, 1.5% in Sepilok East and 1.9% in Paracou, as illustrated in Fig. 5. The mortality rates increased with tree height Fig. 5. The given uncertainty estimates were determined by bootstrapping. A detailed table of the growth and mortality rates can be found in **Extended results**.

DRAFT

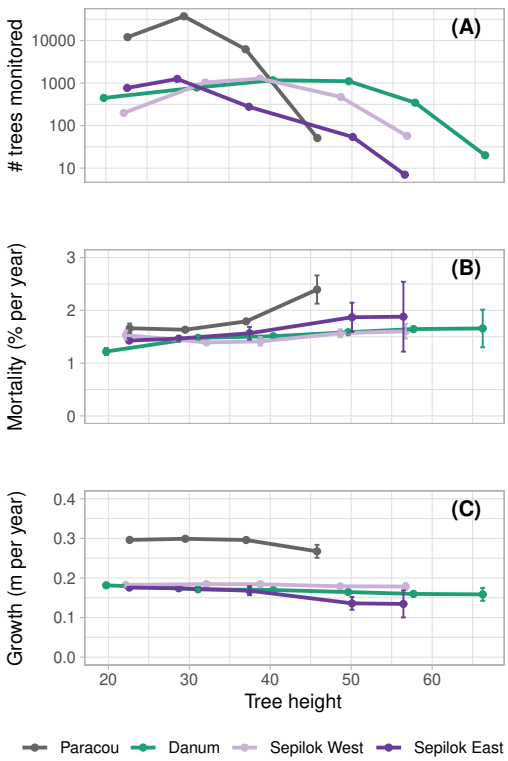

Fig. 5. Plot A shows the distribution of tree heights per site. Plot B shows the mortality rates of trees of different heights in each site, and Plot C gives the growth rate of trees split by height bin. Uncertainty estimates were determined by bootstrapping. The asterisk denotes an uncertainty estimate too large to fit on the plot.

Discussion

Improved tropical crown delineation. Accurately delineating trees in remote sensing data is a long-standing problem in ecology and conservation, and would enable us to efficiently monitor large areas of forests. Detectree2 addresses this problem, delineating individual trees in aerial RGB imagery with high precision and recall. We used Detectree2 to automatically delineate 65,786 trees across three tropical forests. We found that the accuracy of Detectree2 increased with tree height, meaning that that the tall trees which store the most carbon are also the most reliably delineated.

Detectree2 performed well across a range of challenging dense, closed canopy, complex tropical forests. It is able to exactly delineate highly irregular tree crowns within the jigsaw of the canopy rather than simply identifying a bounding box enclosing a tree. This opens up new opportunities for tracking dynamic forest processes including growth and demographics (as demonstrated here) as well as phonological dynamics (where bounding boxes would risk mixing signals). Furthermore, Detectree2 is relatively accessible since it requires a low number of manually delineated trees as training data compared to other methods (30, 38). These advantages are partly due to the fact that Detectree2 is based on a state-of-the-art pre-trained model from the Detectron2 library. Our results did not match the performance reported in (38) but this study is based on semi-synthetic images (i.e. constructed by stitching together existing images) of forests and so is not directly comparable.

Generalisability across sites. There was no obvious relationship between the amount of training data available at a site and the accuracy of automatic delineations attained there. Rather, forest type and tree height distribution seemed to be the key factors for determining forest height. The well differentiated forest at Sepilok West and Danum were the easiest to delineate while the lowest accuracy was in Paracou which has little variation in the height of the visible canopy. Furthermore, at Paracou it is common to observe crowns mixing and growing into each other which makes visually separating the crowns challenging. This in turn is down to soil type and other biogeographic factors.

We found that the accuracy dropped when transferring a model trained on one forest type to predict on another. However, we found that Detectree2 can be quickly trained to perform well on new areas of forest using around 10 images (each ~1 ha in scale) with all visible tree crowns manually delineated. This manual delineation represents approximately 4 hours work. The best performing models were those that were exposed to training data from all the sites and then "honed" with a limited number of training iterations on the site to be predicted on. This suggests that our trained models (provided freely with the Python package[7]) could be transferred to a new site with very little manual data or training iterations. By building a broader database of delineated tree crowns it may be possible to further boost the performance.

We note that the manual delineation was done by different people focusing on separate sites. We were therefore unable to test for the effect of different delineators on our results as they are confounded with site differences. This situation is representative of the expected applications where users with a particular study area will perform the manual delineation for that area. There may be opportunities to crowd-source training data and so compare different users' interpretations of the same image and their effects on the results, but this is beyond the scope of the current study.

Application: Growth and mortality rates. Tall trees store the majority of forest carbon and dominate many important forest nutrient cycles. However, they are rare and therefore poorly represented in traditional field inventories (45) which makes estimating their growth and mortality rates particularly challenging (41–44). Tall trees are also particularly sensitive to the effects of climate change (21), such as increased wind and storms, and as such tracking their dynamics over time is increasingly important. Recent remote sensing studies are bringing new insights into disturbance patterns by mapping the gaps left in the forest canopy after a tree (or multiple trees or branches) have fallen (59–61). Tracking individual trees over time instead of gaps will make it easier to interpret our results in an ecological context and also to compare the results more directly to the available field inventory data.

We demonstrate the value of Detectree2 in forest ecology by combining it with repeat lidar data to estimate growth and mortality rates of 65,786 trees over 14 km^2 of forest. This area is comparable to the total area of all tropical for-

[7] https://github.com/PatBall1/Detectree2

DRAFT

est field plots worldwide (~35 km^2 in the main plot networks (14, 15)). This illustrates the utility of our method for large scale mapping of tall trees. Importantly, uncertainties due to the crown segmentation were assessed independently, and are therefore not conflated with uncertainty in growth and mortality rates.

We found substantially higher growth and mortality rates in French Guiana as compared to the sites in Malaysia. This suggests that these forests are more dynamic, with the increased growth partly balanced by increased mortality. This difference in height growth rate is more than would would be expected from comparing the relative DBH growth rates as measured in the field plot inventories. Further analysis would help to understand the exceptional observed height change at Paracou. Across all sites we found that taller trees had higher mortality rates and lower growth rates. This aligns with large scale analyses of field-based studies (41) and demonstrates the usefulness of tracking tall trees in this way.

We note two important caveats in our analysis of growth and mortality rates. Firstly, we defined mortality as a drop in height of more than a statistically determined threshold. We cannot therefore verify that the tree has died, although it is likely that it has snapped or uprooted. Finally, the fact that our predictions are less accurate for shorter trees may bias the observed trends in vertical tree growth and mortality. Nevertheless, we believe this example application demonstrates the utility of Detectree2 in expanding the sample of trees under observation.

Future methodological developments and applications. As shown in Fig. 2, Detectree2 performs impressively when delineating tall trees. However, it fails to delineate a significant proportion of trees. There is considerable scope to increase the quantity and variety of training data for Mask R-CNN by labelling more trees by hand. A more robust handing of shadowed regions may also support the detection of trees otherwise obscured by their neighbours.

The fact that Detectree2 can be quickly trained to perform well on a new type of forest and imagery demonstrates that it is a useful tool for forest management. Many conservation or restoration projects have access to low-cost imagery from drones or satellites. Detectree2 would allow them to quickly quantify and track the number and size distribution of trees across an entire landscape. In combination with other remote sensing data sources, this could allow for improved carbon stock and dynamics estimation. Estimating carbon stocks in forests has traditionally been done using area-based methods which discard considerable granular information at the individual tree level (19).

We focused on aerial RGB imagery which is the cheapest and most widely available imaging source for tropical forests. We also benefited from the variety of pre-trained models that come with this data type. However, different data sources may provide addition information that would help to discern differences between crowns. In particular, multi-spectral imagery that typically includes additional bands in the near infrared is commonly used to study differences between trees due to the optical properties of vegetation (62). Alternatively, the canopy surface (a raster expressing the height of the canopy) is commonly used in traditional segmentation techniques (e.g. watershed algorithms) and is produced with photogrammetary as a step in generating an orthomosaic. Including this as a layer would add an addition dimension of information that could help to distinguish fine differences in structure. It would be straight forward to include additional (or different) bands to the Detectree2 framework but it would forego the utility of the pre-trained models. Therefore, it is likely that significantly more training data and computational resources would be required to train a model (from scratch) to the desired performance.

Ideally, we could apply this approach to satellite imagery to perform global analyses. Preliminary tests suggest that Detectree2 can accurately delineate trees in RGB imagery at 2 m resolution (see Sup. Note 6) which is equivalent to modern high resolution satellite imagery. If this proves possible it will help answer many long-standing questions in forest ecology as well as providing an important tool for forest management. A "random resizing" augmentation step would further help improve generalisability across resolutions and incorporating "small object" detection features (63) would improve the sensitivity to shorter trees.

While we studied the delineation of a single class (*tree*), Detectree2 is capable of being trained and making predictions on multiple classes. This may allow for low cost species identification and mapping. It may also help to automatically assess liana infestation occurrence. Previously, hyperspectral data has been employed to address this problem but with limited success due to the phylogenetic and spectral diversity of lianas and relatively low spatial resolution of hyperspectral imagery (64, 65). The availablility of Detectree2 as a readily installed, open source Python package means other research groups can test its efficacy on their own research questions.

Conclusions

In this study, we introduced and evaluated Detectree2, a deep learning computer vision method for delineating individual tree crowns. Our model shows considerable skill in complex forest types, particularly in identifying the tallest trees in the canopy.

We combined Detectree2 with repeat airborne lidar data for sites in Malaysia and French Guiana to extract information tree growth and mortality. This enabled us to track around 65,786 trees over an area of 14 km^2. We found that tree growth decreases and mortality increases with increasing tree height, which aligns with previous field studies.

Initial tests suggest that Detectree2 could be scaled to delineate trees in high resolution satellite data, which would enable global scale monitoring of forests at the individual tree level. We hope that our open source tool will be taken up by others and applied to a range of forest ecology questions across different forest types around the world.

ACKNOWLEDGEMENTS

We thank the NERC Earth Observation Data Acquisition and Analysis Service (NEODAAS) for supplying data and computational resources for this study. We thank CEDA for the 2014 Sepilok and Danum remote sensing data. SH received funding from the Centre for Doctoral Training in Application of Artificial Intelligence

DRAFT

to the study of Environmental Risks (AI4ER, EP/S022961/1), which is supported by the Engineering and Physical Sciences Research Council (EPSRC). JGCB was supported by the NERC C-CLEAR doctoral training programme. TDJ and DAC were supported by NERC grant [NE/S010750/1]. DAC was supported by the Franklinia Foundation. We thank Stephen Goult for contributing to discussions. Data collection in French Guiana was supported by CNES who funded the 2016 hyperspectral, RGB and lidar data over Paracou and Labex CEBA (ANR-10-LABX-25) for contributing financial resource for the field validation of manual crown segmentations. The 2019 data in Paracou and 2020 data in Sabah were funded by NERC [NE/S010750/1]. The 2014 Sabah data were also funded by NERC [NE/K016377/1].

Bibliography

1. Yude Pan, Richard A. Birdsey, Jingyun Fang, et al. A large and persistent carbon sink in the world's forests. *Science*, 333(6045):988–993, 2011. doi: 10.1126/science.1201609.
2. R J W Brienen, O L Phillips, T R Feldpausch, et al. Long-term decline of the amazon carbon sink. *Nature*, 519:344–348, 2015. ISSN 1476-4687. doi: 10.1038/nature14283.
3. Wannes Hubau, Simon L Lewis, Oliver L Phillips, et al. Asynchronous carbon sink saturation in african and amazonian tropical forests. *Nature*, 579:80–87, 2020. ISSN 1476-4687. doi: 10.1038/s41586-020-2035-0.
4. Yuanfang Chai, Guilherme Martins, Carlos Nobre, et al. Constraining amazonian land surface temperature sensitivity to precipitation and the probability of forest dieback. *npj Climate and Atmospheric Science*, 4:6, 2021. ISSN 2397-3722. doi: 10.1038/s41612-021-00162-1.
5. Rosie A. Fisher, Charles D. Koven, William R. L. Anderegg, et al. Vegetation demographics in earth system models: A review of progress and priorities. *Global Change Biology*, 24(1): 35–54, 2018. doi: https://doi.org/10.1111/gcb.13910.
6. Charles D Koven, Ryan G Knox, Rosie A Fisher, et al. Benchmarking and parameter sensitivity of physiological and vegetation dynamics using the functionally assembled terrestrial ecosystem simulator (fates) at barro colorado island, panama. *Biogeosciences*, 17(11): 3017–3044, 2020.
7. Sarah M. Gallup, Ian T. Baker, John L. Gallup, et al. Accurate simulation of both sensitivity and variability for amazonian photosynthesis: Is it too much to ask? *Journal of Advances in Modeling Earth Systems*, 13(8):e2021MS002555, 2021. doi: https://doi.org/10.1029/2021MS002555. e2021MS002555 2021MS002555.
8. Natalia Restrepo-Coupe, Loren P. Albert, Marcos Longo, et al. Understanding water and energy fluxes in the amazonia: Lessons from an observation-model intercomparison. *Global Change Biology*, 27(9):1802–1819, 2021. doi: https://doi.org/10.1111/gcb.15555.
9. James R. Kellner, Loren P. Albert, John T. Burley, and K. C. Cushman. The case for remote sensing of individual plants. *American Journal of Botany*, 106(9):1139–1142, 2019. doi: https://doi.org/10.1002/ajb2.1347.
10. Camille Piponiot, Kristina J. Anderson-Teixeira, Stuart J. Davies, et al. Distribution of biomass dynamics in relation to tree size in forests across the world. *New Phytologist*, 234(5):1664–1677, 2022. doi: https://doi.org/10.1111/nph.17995.
11. Pieter A. Zuidema and Peter van der Sleen. Seeing the forest through the trees: how tree-level measurements can help understand forest dynamics. *New Phytologist*, 234(5): 1544–1546, 2022. doi: https://doi.org/10.1111/nph.18144.
12. Michele Dalponte and David A. Coomes. Tree-centric mapping of forest carbon density from airborne laser scanning and hyperspectral data. *Methods in Ecology and Evolution*, 7(10): 1236–1245, 2016. doi: https://doi.org/10.1111/2041-210X.12575.
13. Jérôme Chave, Stuart J Davies, Oliver L Phillips, et al. Ground data are essential for biomass remote sensing missions. *Surveys in Geophysics*, 40:863–880, 2019. ISSN 1573-0956. doi: 10.1007/s10712-019-09528-w.
14. Stuart J. Davies, Iveren Abiem, Kamariah Abu Salim, et al. Forestgeo: Understanding forest diversity and dynamics through a global observatory network. *Biological Conservation*, 253: 108907, 2021. ISSN 0006-3207. doi: https://doi.org/10.1016/j.biocon.2020.108907.
15. ForestPlots.net, Cecilia Blundo, Julieta Carilla, et al. Taking the pulse of earth's tropical forests using networks of highly distributed plots. *Biological Conservation,* 260:108849, 2021. ISSN 0006-3207. doi: https://doi.org/10.1016/j.biocon.2020.108849.
16. David C. Marvin, Gregory P. Asner, David E. Knapp, et al. Amazonian landscapes and the bias in field studies of forest structure and biomass. *Proceedings of the National Academy of Sciences*, 111(48):E5224–E5232, 2014. doi: 10.1073/pnas.1412999111.
17. Victoria Meakem, Alan J. Tepley, Erika B. Gonzalez-Akre, et al. Role of tree size in moist tropical forest carbon cycling and water deficit responses. *New Phytologist*, 219(3):947–958, 2018. doi: https://doi.org/10.1111/nph.14633.
18. James A Lutz, Tucker J Furniss, Daniel J Johnson, et al. Global importance of large-diameter trees. *Global Ecology and Biogeography*, 27(7):849–864, 2018.
19. David A Coomes, Michele Dalponte, Tommaso Jucker, et al. Area-based vs tree-centric approaches to mapping forest carbon in southeast asian forests from airborne laser scanning data. *Remote Sensing of Environment*, 194:77–88, 2017.
20. Atticus E L Stovall, Herman Shugart, and Xi Yang. Tree height explains mortality risk during an intense drought. *Nature Communications*, 10:4385, 2019. ISSN 2041-1723. doi: 10.1038/s41467-019-12380-6.
21. Evan M Gora and Adriane Esquivel-Muelbert. Implications of size-dependent tree mortality for tropical forest carbon dynamics. *Nature Plants*, 7:384–391, 2021. ISSN 2055-0278. doi: 10.1038/s41477-021-00879-0.
22. IPCC. *Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge University Press, 2021.
23. Zhen Zhen, Lindi J. Quackenbush, and Lianjun Zhang. Trends in automatic individual tree crown detection and delineation—evolution of lidar data. *Remote Sensing*, 8(4), 2016. ISSN 2072-4292. doi: 10.3390/rs8040333.
24. Jack H. Hastings, Scott V. Ollinger, Andrew P. Ouimette, et al. Tree species traits determine the success of lidar-based crown mapping in a mixed temperate forest. *Remote Sensing*, 12(2), 2020. ISSN 2072-4292. doi: 10.3390/rs12020309.
25. Mélaine Aubry-Kientz, Raphaël Dutrieux, Antonio Ferraz, et al. A comparative assessment of the performance of individual tree crowns delineation algorithms from als data in tropical forests. *Remote Sensing*, 11(9), 2019. ISSN 2072-4292. doi: 10.3390/rs11091086.
26. Mélaine Aubry-Kientz, Anthony Laybros, Ben Weinstein, et al. Multisensor data fusion for improved segmentation of individual tree crowns in dense tropical forests. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:3927–3936, 2021. doi: 10.1109/JSTARS.2021.3069159.
27. Jakob Iglhaut, Carlos Cabo, Stefano Puliti, et al. Structure from motion photogrammetry in forestry: a review. *Current Forestry Reports*, 5:155–168, 2019. ISSN 2198-6436. doi: 10.1007/s40725-019-00094-3.
28. André Almeida, Fabio Gonçalves, Gilson Silva, et al. Individual tree detection and qualitative inventory of a eucalyptus sp. stand using uav photogrammetry data. *Remote Sensing*, 13 (18), 2021. ISSN 2072-4292. doi: 10.3390/rs13183655.
29. Anderson Aparecido dos Santos, Jose Marcato Junior, Márcio Santos Araújo, et al. Assessment of cnn-based methods for individual tree detection on images captured by rgb cameras attached to uavs. *Sensors*, 19(16):3595, 2019.
30. Ben G Weinstein, Sergio Marconi, Stephanie Bohlman, et al. Individual tree-crown detection in rgb imagery using semi-supervised deep learning neural networks. *Remote Sensing*, 11 (11):1309, 2019.
31. Ben G Weinstein, Sergio Marconi, Stephanie A Bohlman, et al. A remote sensing derived data set of 100 million individual tree crowns for the national ecological observatory network. *Elife*, 10:e62922, 2021.
32. Teja Kattenborn, Jens Leitloff, Felix Schiefer, and Stefan Hinz. Review on convolutional neural networks (cnn) in vegetation remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing*, 173:24–49, 2021. ISSN 0924-2716. doi: https://doi.org/10.1016/j.isprsjprs.2020.12.010.
33. Xiao Xiang Zhu, Devis Tuia, Lichao Mou, et al. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*, 5(4):8–36, 2017. doi: 10.1109/MGRS.2017.2762307.
34. Kunyong Yu, Zhenbang Hao, Christopher J. Post, et al. Comparison of classical methods and mask r-cnn for automatic tree detection and mapping using uav imagery. *Remote Sensing*, 14(2), 2022. ISSN 2072-4292. doi: 10.3390/rs14020295.
35. Zhenbang Hao, Lili Lin, Christopher J Post, et al. Automated tree-crown and height detection in a young forest plantation using mask region-based convolutional neural network (mask r-cnn). *ISPRS Journal of Photogrammetry and Remote Sensing*, 178:112–123, 2021.
36. Nuri Erkin Ocer, Gordana Kaplan, Firat Erdem, et al. Tree extraction from multi-scale uav images using mask r-cnn with fpn. *Remote Sensing Letters*, 11(9):847–856, 2020. doi: 10.1080/2150704X.2020.1784491.
37. Gensheng Hu, Tongxiang Wang, Mingzhu Wan, et al. Uav remote sensing monitoring of pine forest diseases based on improved mask r-cnn. *International Journal of Remote Sensing*, 43(4):1274–1305, 2022. doi: 10.1080/01431161.2022.2032455.
38. José R G Braga, Vinícius Peripato, Ricardo Dalagnol, et al. Tree crown delineation algorithm based on a convolutional neural network. *Remote Sensing*, 12(8):1288, 2020.
39. Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.
40. Yuxin Wu, Alexander Kirillov, Francisco Massa, et al. Detectron2. https://github.com/facebookresearch/detectron2, 2019.
41. Yoshiko Iida, Lourens Poorter, Frank Sterck, et al. Linking size-dependent growth and mortality with architectural traits across 145 co-occurring tropical tree species. *Ecology*, 95(2): 353–363, 2014.
42. Sarah J Richardson, Mark C Smale, Jennifer M Hurst, et al. Large-tree growth and mortality rates in forests of the central north island, new zealand. *New Zealand Journal of Ecology*, pages 208–215, 2009.
43. David A Coomes, Richard P Duncan, Robert B Allen, and James Truscott. Disturbances prevent stem size-density distributions in natural forests from following scaling relationships. *Ecology letters*, 6(11):980–989, 2003.
44. Helene C Muller-Landau, Richard S Condit, Jerome Chave, et al. Testing metabolic ecology theory for allometric scaling of tree size, growth and mortality in tropical forests. *Ecology letters*, 9(5):575–588, 2006.
45. Jennifer M Hurst, Robert B Allen, David A Coomes, and Richard P Duncan. Size-specific tree mortality varies with neighbourhood crowding and disturbance in a montane nothofagus forest. *PloS one*, 6(10):e26670, 2011.
46. Alexander Shenkin, Chris J Chandler, Doreen S Boyd, et al. The world's tallest tropical tree in three dimensions. *Frontiers in Forests and Global Change*, 2:32, 2019.
47. Reuben Nilus, Colin R Maycock, N Majalap-Lee, and DFRP Burslem. Nutrient limitation of tree seedling growth in three soil types found in sabah. *Journal of Tropical Forest Science*, pages 133–142, 2011.
48. Sylvie Gourlet-Fleury, Jean-Marc Guehl, and Olivier Laroussinie. *Ecology and management of a neotropical rainforest: lessons drawn from Paracou, a long-term experimental research site in French Guiana*. Elsevier, 2004.
49. Fabien Wagner, Bruno Hérault, Clément Stahl, et al. Modeling water availability for trees in tropical forests. *Agricultural and Forest Meteorology*, 151(9):1202–1213, 2011. ISSN 0168-1923. doi: https://doi.org/10.1016/j.agrformet.2011.04.012.
50. AgiSoft. Agisoft metashape (version 1.7.3), 2021.
51. Matthew J Westoby, James Brasington, Niel F Glasser, et al. 'structure-from-motion'photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology*, 179:300–314, 2012.
52. Teja Kattenborn, Felix Schiefer, Julian Frey, et al. Spatially autocorrelated training and validation samples inflate performance assessment of convolutional neural networks. *ISPRS Open Journal of Photogrammetry and Remote Sensing*, 5:100018, 2022. ISSN 2667-3932. doi: https://doi.org/10.1016/j.ophoto.2022.100018.
53. Tsung-Yi Lin, Michael Maire, Serge Belongie, et al. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
54. Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016.
55. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for

DRAFT

image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

56. Tsung-Yi Lin, Piotr Dollár, Ross Girshick, et al. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.
57. Jean-Yves Audibert and Olivier Catoni. Robust linear least squares regression. *The Annals of Statistics*, 39(5):2766–2794, 2011.
58. Jean-Romain Roussel, John Caspersen, Martin Béland, et al. Removing bias from lidar-based estimates of canopy height: Accounting for the effects of pulse density and footprint size. *Remote Sensing of Environment*, 198:1–16, 2017.
59. Raquel Fernandes Araujo, Samuel Grubinger, Carlos Henrique Souza Celes, et al. Strong temporal variation in treefall and branchfall rates in a tropical forest is related to extreme rainfall: results from 5 years of monthly drone data for a 50 ha plot. *Biogeosciences*, 18(24): 6517–6531, 2021.
60. KC Cushman, Matteo Detto, Milton García, and Helene C Muller-Landau. Soils and topography control natural disturbance rates and thereby forest structure in a lowland tropical landscape. *Ecology Letters*, 2022.
61. Claudia Huertas, Daniel Sabatier, Géraldine Derroire, et al. Mapping tree mortality rate in a tropical moist forest using multi-temporal lidar. *International Journal of Applied Earth Observation and Geoinformation*, 109:102780, 2022. ISSN 1569-8432. doi: https://doi.org/10.1016/j.jag.2022.102780.
62. Edward B. Knipling. Physical and physiological basis for the reflectance of visible and near-infrared radiation from vegetation. *Remote Sensing of Environment*, 1(3):155–159, 1970. ISSN 0034-4257. doi: https://doi.org/10.1016/S0034-4257(70)80021-9.
63. Kang Tong and Yiquan Wu. Deep learning-based detection from the perspective of small or tiny objects: A survey. *Image and Vision Computing*, 123:104471, 2022. ISSN 0262-8856. doi: https://doi.org/10.1016/j.imavis.2022.104471.
64. Mathias Wessel, Melanie Brandmeier, and Dirk Tiede. Evaluation of different machine learning algorithms for scalable classification of tree types and tree species based on sentinel-2 data. *Remote Sensing*, 10(9):1419, 2018.
65. Ewa Grabska, David Frantz, and Katarzyna Ostapowicz. Evaluation of machine learning algorithms for forest stand species mapping using sentinel-2 imagery and environmental data in the polish carpathians. *Remote Sensing of Environment*, 251:112103, 2020.
66. QGIS Association. Qgis geographic information system, 2021.
67. Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.
68. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.
69. Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

DRAFT

Word Counts

This section is *not* included in the word count.

Notes on Methods in Ecology and Evolution.

- Abstract: 350 words and should list the main results and conclusions, using FOUR simple, factual, numbered statements
- Main text: no more than 7000-8000 words for Standard Articles, 3000-4000 words for Applications and Practical Tools

Statistics on word count.

```
!!! File Article.bbl not found in path []. !!!
File: Article.tex
Encoding: utf8
Sum count: 6752
Words in text: 6180
Words in headers: 85
Words outside text (captions, etc.): 479
Number of headers: 25
Number of floats/tables/figures: 7
Number of math inlines: 7
Number of math displayed: 1
Subcounts:
  text+headers+captions (#headers/#floats/#inlines/#displayed)
  108+14+0 (1/0/0/0) _top_
  343+1+0 (1/0/0/0) Abstract
  26+0+0 (0/0/0/0) main
  900+1+4 (1/0/0/0) Section: Introduction
  0+3+0 (1/0/0/0) Section: Materials and Methods
  185+2+0 (1/0/0/0) Subsection: Study sites
  144+3+32 (1/1/0/0) Subsection: Remote sensing data
  204+4+79 (1/1/1/0) Subsection: Manual tree crown data
  205+2+0 (1/0/0/0) Subsection: Data preparation} \label{methods_prep
  391+4+24 (1/0/0/0) Subsection: Model architecture and parameterisation
  355+4+9 (1/0/1/0) Subsection: Training and model selection} \label{methods_training
  60+3+0 (1/0/0/0) Subsection: Transferability across sites
  419+2+0 (1/0/5/1) Subsection: Performance evaluation
  275+6+0 (1/0/0/0) Subsection: Application to monitoring growth and mortality
  54+1+0 (1/0/0/0) Subsection: Computation
  0+1+0 (1/0/0/0) Section: Results
  194+6+86 (1/2/0/0) Subsection: Performance by site and tree height
  197+4+123 (1/1/0/0) Subsection: Performance between forest types
  271+4+118 (1/2/0/0) Subsection: Application: Growth and mortality
  0+1+0 (1/0/0/0) Section: Discussion
  227+4+0 (1/0/0/0) Subsection: Improved tropical crown delineation
  338+3+4 (1/0/0/0) Subsection: Generalisability across sites
  422+5+0 (1/0/0/0) Subsection: Application: Growth and mortality rates
  539+5+0 (1/0/0/0) Subsection: Future methodological developments and applications
  322+1+0 (1/0/0/0) Section: Conclusions
  1+1+0 (1/0/0/0) Section: Bibliography

(errors:1)
```

DRAFT

Supplementary Note 1: Study sites and remote sensing data collection

A. Sepilok (East & West) and Danum. Sepilok Forest Reserve (5°50' N 117°56' E) is a region of lowland tropical rainforest in Sabah, a state in Malaysia. The Reserve is one of the oldest protected areas of forest in Asia, being founded by the Sabah Forest Department in 1931. The Reserve spans nearly 4500 ha, with ground elevation varying between 50 and 250 metres above sea level. Three distinct forest types are present: alluvial dipterocarp, sandstone dipterocarp and heath forest. Danum Valley contains lowland tropical rain forests dominated by dipterocarps and are among the tallest forests on the planet (46). The three Malaysian sites experience a similar climate with approximately 2300 mm rainfall per year with the wettest months between November and February (47).

The 2014 lidar data of Sepilok and Danum were collected using a Leica ALS50-II ALS flown at an altitude of nearly 2000 metres, attached to the belly of a Dornier 228-201. The ALS sensor emitted pulses at around 80 Hz with a field of view of 12.0°, and a footprint of about 40 cm diameter. The average pulse density was 7.3 m-2. The ALS data were preprocessed by NERC's Data Analysis Node and delivered in LAS format. The 2020 lidar data in Sepilok was collected using a Riegl LMS-Q780 mounted on a helicopter flying at 250 metres altitude at a ground speed of approximately 100 km/hr. This resulted in an average pulse density of 16 m^{-2}. Further processing used LAStools (http://rapidlasso.com/lastools/). Points were split into two groups, ground and non-ground, and a digital elevation model (DEM) was fitted to the ground points, producing a raster of 0.5 m resolution. The DEM elevations were subtracted from elevations of all non-ground returns, known as the digital surface model (DSM) to create a canopy height model (CHM) raster of resolution 0.5 m.

The 2014 RGB imagery was collected using a Leica RCD105 Digital Camera, attached to a plane flown at an altitude of 796 metres, with a ground resolution of 10 cm. Photographs have been orthorectified and collated into homogenous mosaics using Agisoft Metashape. This software used the Structure from Motion method to calculate the elevation of the observed surface, of which the photographs are mapped to, This results in georeferenced photographs that are corrected for distortions. The files were delivered in the GeoTIFF format.

B. Paracou. The Paracou field station is situated in a lowland tropical forest near Sinnamary, French Guiana (5°16'N 52 °55'W). It is a forest similar to Sepilok Forest Reserve, which is also a lowland tropical forest.

ALS data were acquired in September 2016, and November 2019 by ALTOA, operating a RIEGL LMS-Q780 sensor attached to an aircraft flying at 800m. On all dates, the scan frequency was 400 kHz and the final point density was above 50 m^{-2}. The creation of a CHM was carried out using the same process as used for Sepilok.

On the same flights as the ALS scans, RGB images were collected using an IXA180 Phase One camera with an 8 cm ground sampling distance.

Hyperspectral imagery was also collected over Paracou, with a Hyspex VNIR-1600 (Hyspex NEO, Skedsmokorset, Norway) sensor-mounted alongside the RIEGL scanner. Its bands covered a spectral range of 414–994 nm with a spectral sampling distance of 3.64 nm. Images were orthorectified and georeferenced to 1 m spatial resolution with the PARGE software using the DSM from the lidar data.

DRAFT

Supplementary Note 2: Tree crown data

A. Manual delineation. Manual tree crown labelling was carried out at each study site, to create training and test sets for our analysis. Labelling was carried out in open-source graphical geospatial software, QGIS (66), using a combination of RGB, lidar, and, in the case of Paracou, hyperspectral imagery. To increase the contrast between individual trees the colours of the RGB imagery were stretched. Polygons were carefully drawn around the perimeter of all distinguishable trees for a number of areas in each region. JGCB carried out the manual labelling for Paracou. A visualisation of the difference between the original and the stretched image is given in Fig. S1.

B. Training and validation data. Examples of the training data are given in Fig. S4.

Supplementary Note 3: Data preparation and processing

To apply Detectree2 to remote sensing RGB images of tropical forests, the geospatial raster image must be tiled and converted into a png format, and the manual crown segmentations used for training must also be converted from geospatial shapefiles into JSON files that align with the pngs. These training tiles are ingested into Detectree2, which then learns how to identify a tree, and then can make predictions on new tiles.

Training tiles were separated into 5-folds so that cross validation could be performed during the hyperparameter tuning phase. Testing tiles were spatially separated from the training tiles and only seen by the network at the evaluation phase.

Fig. S1. An original RGB image is displayed on the left, while a stretched RGB image is displayed on the right of this figure. Stretching the colours of the image to the values allows for easier identification of individual trees when carrying out manual tree crown delineations. The effect is particularly noticeable in the lower left corner of the images.

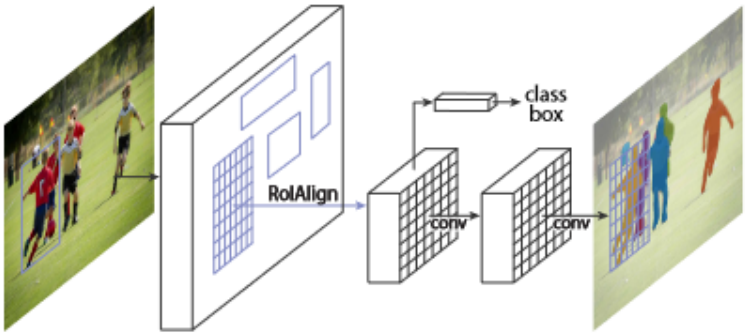

Fig. S2. Illustration of the Mask R-CNN architecture from He et al. (39).

Supplementary Note 4: Detectron2 library and Mask R-CNN framework

A. Model architecture. Mask R-CNN (39) is a framework for instance segmentation. It builds on Faster R-CNN (67), which carries out object detection using a Region Proposal Network (RPN). An RPN is a fully convolutional network, trained end-to-end, that generates Regions of Interest (RoIs) for each image. These RoIs have object bounds and objectness scores attached to them, giving the bounds of the RoI, and the likelihood of the RoI containing an object. These RoIs are then passed through fully connected convolutional networks to determine the class of image contained, and the exact mask of each object, within the respective bounding box. For full details on the structure of Mask R-CNN, please refer to the GitHub repository for this work (https://github.com/PatBall1/Detectree2). The schematic of the architecture of the model architecture is given in Fig. S2.

We selected the R101-FPN configuration - this architecture combines a 101 layer deep ResNet (68) module with a Feature Pyramid Network module.

This model is an improvement upon existing architectures, such as architectures performing object detection. An example of the different predictions is given in Figure S3.

B. Data augmentation. The training data was augmented by submitting the training data to a variety of transformations including vertical and horizontal flips, rotation, and varying the saturation and contrast of the images. These augmentations are designed to increase the variety of training data seen by the model, to allow it to generalise better to new images.

C. Hyperparameter tuning. As with all deep networks, there were several of hyperparameters that controlled the way the networks handled data handling and were trained. The first of these relates to the architecture of the algorithm, namely the number of hidden layers in the ResNet backbone (55) of the network. ResNets are used as the backbone of Mask R-CNN as very deep neural networks are particularly difficult to train, due to the vanishing gradients problem (69). ResNets use skip connections, whereby certain layers in the network are skipped during backpropagation to avoid this problem.

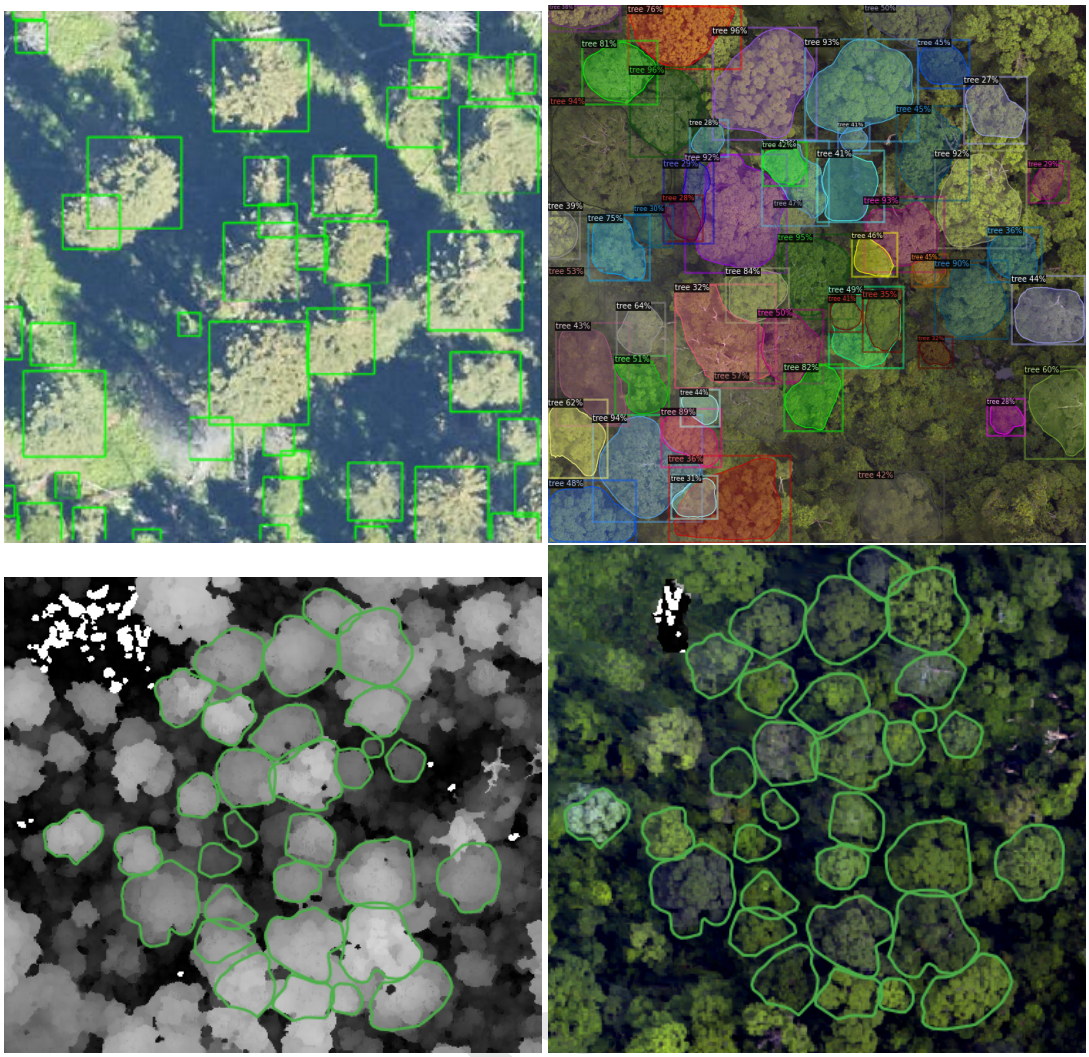

Fig. S3. (A) Crown bounding boxes predicted by DeepForest (Weinstein et al., 2019), and (B) crowns predicted by Detectree2. The colours in plot B merely distinguish predicted trees. A comparison of manually delineated crowns, overlaid on lidar (C) and RGB (D).

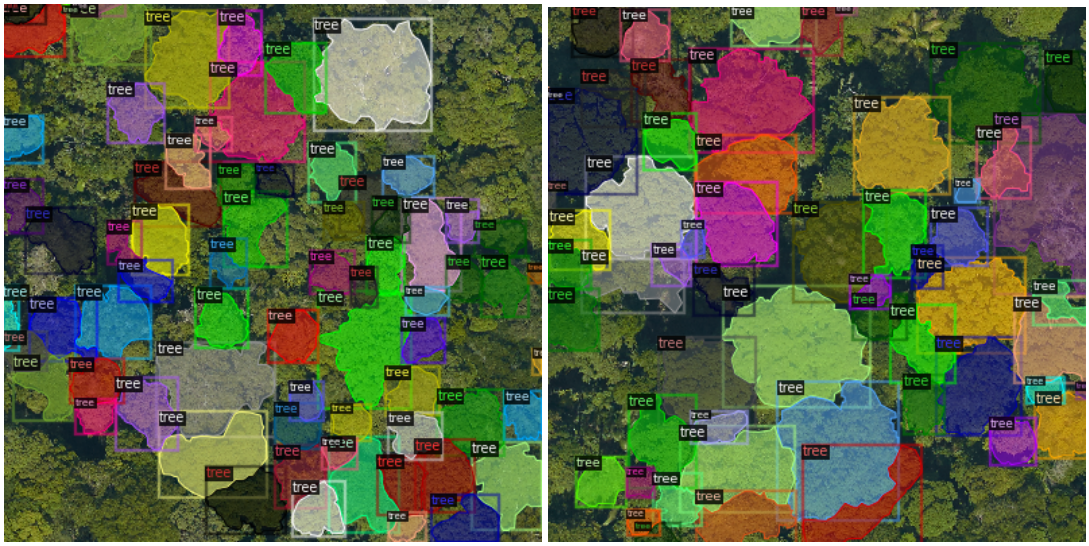

Fig. S4. Examples of training data provided to Mask R-CNN. The different colours help to distinguish between trees.

Table S1. Tunable hyperparameters (with their optimised value) and a description of their purpose.

Hyperparameter (=val)	Description
dl_nums_workers (=2)	To speed up the training process, we make use of the num_workers optional attribute of the DataLoader class. The num_workers attribute tells the data loader instance how many sub-processes to use for data loading. By default, the num_workers value is set to zero, and a value of zero tells the loader to load the data inside the main process. This means that the training process will work sequentially inside the main process. After a batch is used during the training process and another one is needed, we read the batch data from the disk. Now, if we have a worker process, we can make use of the fact that our machine has multiple cores. This means that the next batch can already be loaded and ready to go by the time the main process is ready for another batch. This is where the speed up comes from. The batches are loaded using additional worker processes and are queued up in memory.
ims_per_batch (=2)	If we use 16 GPUs and IMS_PER_BATCH = 32, each GPU will see 2 images per batch.
gamma (=0.1)	The iteration number to decrease the learning rate by GAMMA.
backbone_freeze_at (=3)	Freeze the first several stages so they are not trained. There are 5 stages in ResNet. The first is a convolution, and the following stages are each group of residual blocks. Freezing a layer prevents its weights from being modified. This technique is often used in transfer learning, where the base model(trained on some other dataset is frozen.
warmup_iters (=120)	Warm-up steps are just a few updates with low learning rate before training. After this warm-up, the regular learning rate is use, (schedule) to train the model to convergence. The idea is that this helps the network to slowly adapt to the data intuitively. If the data set is highly differentiated, it can suffer from a sort of "early over-fitting". If your shuffled data happens to include a cluster of related, strongly-featured observations, your model's initial training can skew badly toward those features - or worse, toward incidental features that are not truly related to the topic at all. Warm-up is a way to reduce the primacy effect of the early training examples. Without it, one may need to run a few extra epochs to get the convergence desired, as the model un-trains those early superstitions. Many models afford this as a command-line option. The learning rate is increased linearly over the warm-up period. If the target learning rate is p and the warm-up period is n, then the first batch iteration uses $1p/n$ for its learning rate; the second uses $2p/n$, and so on: iteration i uses $i * p/n$, until we hit the nominal rate at iteration n. This means that the first iteration gets only $1/n$ of the primacy effect. This does a reasonable job of balancing that influence. Note that the ramp-up is commonly on the order of one epoch – but is occasionally longer for particularly skewed data, or shorter for more homogeneous distributions. One may want to adjust, depending on how functionally extreme ones batches can become when the shuffling algorithm is applied to the training set.
momentum (=0.9)	Momentum in neural networks is a variant of the stochastic gradient descent. It replaces the gradient with a momentum which is an aggregate of gradients. Momentum can increase speed when the cost surface is highly non-spherical because it damps the size of the steps along with directions of high curvature thus yielding a larger effective learning rate along with the directions of low curvature.
batch_size_per_image (=1024)	The batch size defines the number of samples that will be propagated through the network. For instance, let's say you have 1050 training samples and you want to set up a batch_size equal to 100. The algorithm takes the first 100 samples (from 1st to 100th) from the training dataset and trains the network. Next, it takes the second 100 samples (from 101st to 200th) and trains the network again. We can keep doing this procedure until we have propagated all samples through of the network. Problem might happen with the last set of samples. In our example, we have used 1050 which is not divisible by 100 without remainder. The simplest solution is just to get the final 50 samples and train the network. Advantages of using a batch size < number of all samples: It requires less memory. Since you train the network using fewer samples, the overall training procedure requires less memory. That is especially important if you are not able to fit the whole dataset in your machine's memory. Typically networks train faster with mini-batches. That is because we update the weights after each propagation. In our example we have propagated 11 batches (10 of them had 100 samples and 1 had 50 samples) and after each of them we have updated our network's parameters. If we used all samples during propagation we would make only 1 update for the network's parameter. Disadvantages of using a batch size < number of all samples: The smaller the batch the less accurate the estimate of the gradient will be.
weight_decay (=0.001)	Weight decay is a regularization technique by adding a small penalty, usually the L2 norm of the weights (all the weights of the model), to the loss function. loss = loss + weight decay parameter * L2 norm of the weights. Why do we use weight decay? To prevent overfitting. To keep the weights small and avoid exploding gradient. Because the L2 norm of the weights are added to the loss, each iteration of your network will try to optimize/minimize the model weights in addition to the loss.
base_lr (=0.0003389)	In machine learning and statistics, the learning rate is a tuning parameter in an optimization algorithm that determines the step size at each iteration while moving toward a minimum of a loss function. Since it influences to what extent newly acquired information overrides old information, it metaphorically represents the speed at which a machine learning model "learns". In the adaptive control literature, the learning rate is commonly referred to as gain. In setting a learning rate, there is a trade-off between the rate of convergence and overshooting. While the descent direction is usually determined from the gradient of the loss function, the learning rate determines how big a step is taken in that direction. A too high learning rate will make the learning jump over minima but a too low learning rate will either take too long to converge or get stuck in an undesirable local minimum. In order to achieve faster convergence, prevent oscillations and getting stuck in undesirable local minima the learning rate is often varied during training either in accordance with a learning rate schedule or by using an adaptive learning rate.
max_iter (=462)	An iteration describes the number of times a batch of data passed through the algorithm. In the case of neural networks, that means the forward pass and backward pass. So, every time you pass a batch of data through the neural network, you completed an iteration. When you increase the number of iterations, you get closer to the minima and the set of optimal parameters, and these are what improve performance.

DRAFT

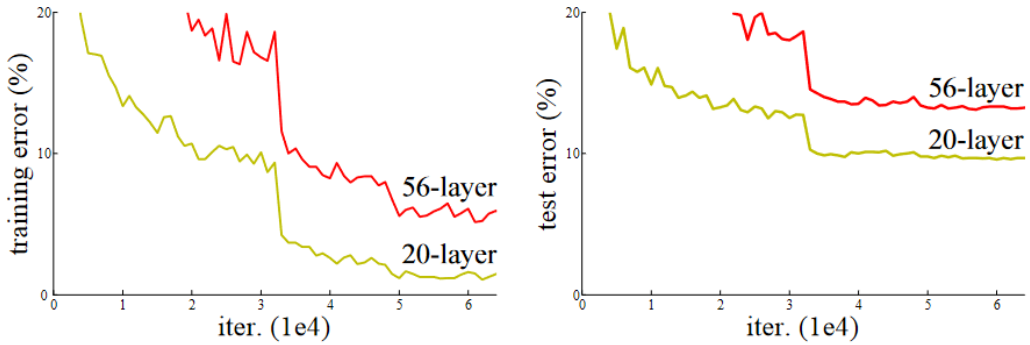

Fig. S5. These plots are taken from He et al., (2016) and they illustrate that deeper neural networks do not necessarily learn as well as shallower neural networks.

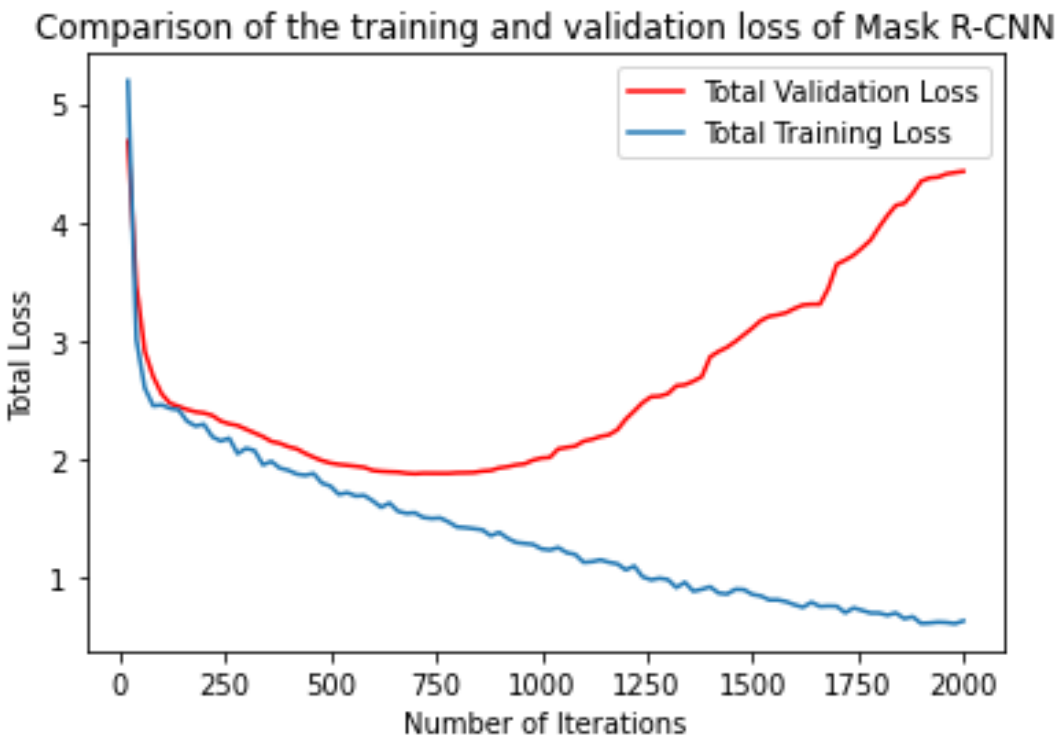

Fig. S6. The total training and validation loss of Mask R-CNN as the model trained. Both total training and validation loss were calculated every 20 iterations.

Other hyperparameters that can be optimised in Mask R-CNN relate to the training of the network. Namely the learning rate, the number of iterations and the batch size. The optimisation of these hyperparameters was carried out using Weights and Biases, which uses grid search to determine the optimal hyperparameters. The hyperparameters selected are given in Table S4.1.

We found that while increasing the depth of the ResNet increased the training time of the algorithm, it improved accuracy scores. The batch size was limited by computing resources available. The number of iterations was the key metric to optimise, as we found that our model began to overfit the training data if trained for too many iterations. To determine the number of iterations, we plotted a graph (Fig. S6) of both the total training loss and the total validation loss to determine the minimum of the total validation loss, and hence the optimal number of iterations to train the model. As Mask R-CNN consists of three predictions, namely the class, bounding box and mask, the total loss is the sum of these three losses.

We see that after around 800 iterations, the model started to overfit the training data, at the expense of performance of the validation dataset. As such we determined to stop the training of Mask R-CNN at 800 iterations, and then used the saved weights after 800 iterations to predict tree crowns in Sepilok.

Supplementary Note 5: Evaluation metrics

Two key metrics are used in this work to evaluate the performance of the model: firstly, the AP50 score, and the F_1 Score. AP50 relates to the area under the precision-recall curve (AUC-PR) evaluated at a particular threshold for the Intersection over Union, in our case 0.5. A high value for AP50 means that we are seeing both high precision and recall of the model at our particular IoU cutoff. The IoU cutoff was selected as 0.5, on the basis of previous studies evaluating the accuracy of individual tree delineation methods. Visual examples of different IoU values are given in Fig. 1.

Supplementary Note 6: Sensitivity to image resolution

We performed a preliminary analysis on how sensitive the delineation accuracy is to changes in the image resolution. We then carried out another test, where we trained our model on the highest resolution available to us (10 cm), before testing it on lower resolution to determine the model's sensitivity to resolution. The lower resolutions were chosen as they are typical resolutions of modern, high-resolution satellite imagery. The final test was to train and test the model on low resolutions to determine if this workflow would increase the skill of the model on lower resolutions. As seen in Fig. S7, illustrate that our model trained on high resolution (0.1 m) imagery is most successful when predicting on similar resolutions, and its performance degrades for resolutions an order of magnitude greater (1 m and 2 m). However, we illustrate that when the model is trained on similar resolutions to those it is tested on, performance is largely maintained. It is only on resolutions greater than 1 m that the performance degrades significantly.

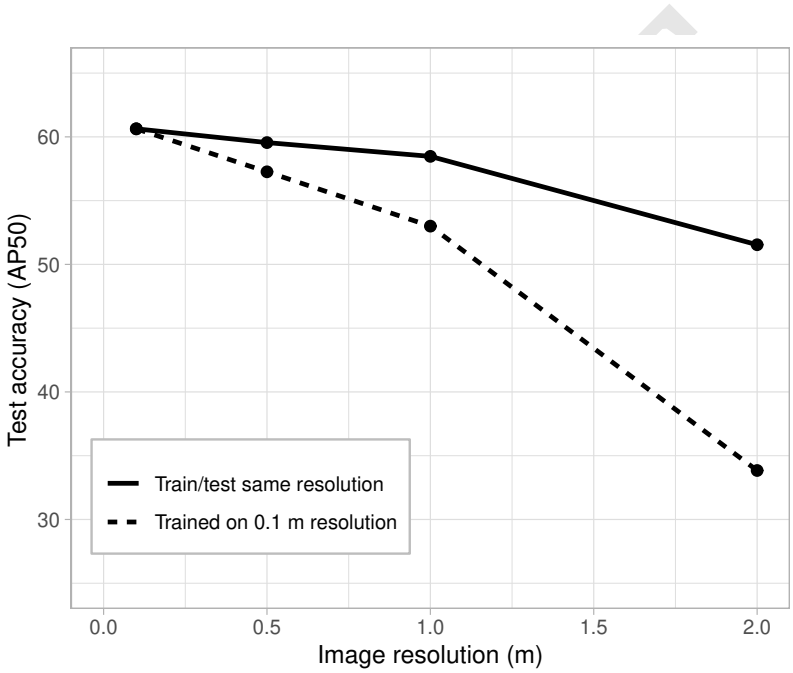

Fig. S7. The sensitivity of the accuracy of the segmentations to the resolution of images used in training and testing.

Supplementary Note 7: Growth and mortality details

Growth and mortality rates were estimated from the CHMs of the repeat lidar data (taken at all sites as detailed in Table 1). Tree height was defined by the median value of the CHM within the overlayed crown polygon. To calculate growth rate (in m / yr), we evaluated the difference in height of individual trees in the two years of measurements, and divided by the time between the measurements (assuming an approximately constant growth rate).

Mortality estimates were derived by fitting a robust least squares regression to the change in tree height against the original tree height. Mortality events were defined as a drop in height of 3 standard deviations or more from this fit. The rate of mortality (%/yr) was calculated as the percentage of detected trees that died divided by the time between scans. The tree mortality rates were grouped into height bins (depending on the original height of each tree). A drop of three standard deviations is strict but it is possible that drops could result from non-mortality events such as snapping or other wind damage. Uncertainty estimates were determined by bootstrapping (repeatedly sampling from the complete set of predicted crowns).

Table S2. The coefficients and intercepts for the robust least squares fit between original tree height and the change in tree height.

Site	Slope	Intercept
Danum	-0.110 ± 0.006	1.90 ± 0.3
Paracou	-0.028 ± 0.0003	1.06 ± 0.01
Sepilok East	-0.060 ± 0.004	1.64 ± 0.11
Sepilok West	-0.072 ± 0.003	1.68 ± 0.12

DRAFT