Received: 23 March 2020 | Accepted: 4 August 2020

DOI: 10.1111/2041-210X.13472

APPLICATION

Methods in Ecology and Evolution BRITISH ECOLOGICAL SOCIETY

DeepForest: A PYTHON package for RGB deep learning tree crown delineation

Ben G. Weinstein[1] | **Sergio Marconi**[1] | **Mélaine Aubry-Kientz**[2] | **Gregoire Vincent**[2] | **Henry Senyondo**[1] | **Ethan P. White**[1]

[1]Department of Wildlife Ecology and Conservation, University of Florida, Gainesville, FL, USA

[2]AMAP, IRD, CNRS, INRA, University of Montpellier, CIRAD, Montpellier, France

Correspondence
Ben G. Weinstein
Email: benweinstein2010@gmail.com

Funding information
Gordon and Betty Moore Foundation, Grant/Award Number: GBMF4563; National Science Foundation, Grant/Award Number: 1926542

Handling Editor: Sydne Record

Abstract

1. Remote sensing of forested landscapes can transform the speed, scale and cost of forest research. The delineation of individual trees in remote sensing images is an essential task in forest analysis. Here we introduce a new PYTHON package, DeepForest that detects individual trees in high resolution RGB imagery using deep learning.
2. While deep learning has proven highly effective in a range of computer vision tasks, it requires large amounts of training data that are typically difficult to obtain in ecological studies. DeepForest overcomes this limitation by including a model pretrained on over 30 million algorithmically generated crowns from 22 forests and fine-tuned using 10,000 hand-labelled crowns from six forests.
3. The package supports the application of this general model to new data, fine tuning the model to new datasets with user labelled crowns, training new models and evaluating model predictions. This simplifies the process of using and retraining deep learning models for a range of forests, sensors and spatial resolutions.
4. We illustrate the workflow of DeepForest using data from the National Ecological Observatory Network, a tropical forest in French Guiana, and street trees from Portland, Oregon.

KEYWORDS

crown delineation, deep learning, forests, NEON, remote sensing, RGB, tree crowns

1 | INTRODUCTION

Airborne individual tree delineation is a central task for forest ecology and the management of forested landscapes. The growth in sensor quality and data availability has raised hopes that airborne tree maps can complement traditional ground-based surveys (Hamraz, Contreras, & Zhang, 2016). Most approaches to tree delineation in remote sensing use three-dimensional LIDAR data (Coomes et al., 2017), which is currently available for only a small fraction of the Earth's surface. In contrast, high resolution RGB data has widespread coverage from commercial and government sources and is readily collected using unmanned aerial vehicles. As a result, there is an increasing need for RGB-based tree delineation approaches with easy to use open-source implementations.

The introduction of deep neural networks has greatly enhanced the performance of remote sensing solutions for detecting objects in geospatial images (Zhu et al., 2017). Deep learning models use a series of hierarchical layers to learn directly from training data instead of using expert designed features. Initial layers learn general representations, such as colours and shapes, and subsequent layers learn specific object representations. There are several barriers to apply deep learning to ecological applications including insufficient technical expertise, a lack of large amounts of training data, and the need for significant computational resources. DeepForest provides easy access to deep learning for tree delineation by creating a simple interface for training object detection models, using them to make predictions, and evaluating the accuracy of those predictions. DeepForest also includes a prebuilt model (based on Weinstein, Marconi, Bohlman, Zare, & White, 2020) pretrained on tens of millions of LiDAR generated crowns and fine-tuned using over 10,000 hand-labelled crowns from diverse forests in the National Ecological Observatory Network. Users can apply this model to detect trees in new imagery or provide additional hand-labelled data to fine-tune performance for a specific site or forest type. Predictions from the model for an average 1 km^2 tile can be made in 7 min on a single CPU and DeepForest has built-in support for running on GPU resources to dramatically increase the speed of prediction at large scales.

2 | DEEPFOREST SOFTWARE

DeepForest is an open source (MIT license) Python package supporting Python 3.6 and Python 3.7 and has been tested on Windows, macO, and Linux operating systems. It can be installed using the Python Package Index (https://pypi.org/project/deepforest/) or using the conda package manager for Windows, Linux and OSX (https://github.com/conda-forge/deepforest-feedstock). The software is openly developed on GitHub (https://github.com/weecology/DeepForest) with automated testing and each release is archived on Zenodo (https://doi.org/10.5281/zenodo.2538143). All DeepForest functions are documented online with reproducible examples (https://deepforest.readthedocs.io/) and video tutorials.

2.1 | Prebuilt model

DeepForest currently includes one prebuilt model (available by running deepforest.use_release) that was trained on data from the National Ecological Observatory Network (NEON) using a semi-supervised approach outlined in Weinstein, Marconi, Bohlman, Zare, and White (2019), Weinstein, Marconi, Bohlman, et al. (2020) (Figure 1). The model was pretrained on data from 22 NEON sites using an unsupervized LiDAR based algorithm (Silva et al., 2016) to generate millions of moderate quality annotations for model pretraining. The pretrained model was then retrained based on over 10,000 hand-annotations of RGB imagery from six sites (MLBS, NIWO, OSBS, SJER, TEAK, LENO; see NEON site abbreviations Supporting Information S1). The full workflow is shown in Figure 1. While LIDAR data is used to facilitate data generation for the prebuilt model, prediction relies only on RGB data, allowing the model to be used to detect trees using RGB imagery alone. This prebuilt model extends the methods from Weinstein et al. (2019), Weinstein, Marconi, Bohlman, et al. (2020) by pretraining on a much larger number of trees (30 million LIDAR-generated crowns compared to 10 million in Weinstein, Marconi, Bohlman, et al., 2020) and diversity of sites (22 instead of 4 in Weinstein et al., 2020). Additional details on the modeling approach, data generation and model evaluation are available in Weinstein et al. (2019), Weinstein, Marconi, Bohlman, et al. (2020) and a brief summary lis provided in Supporting Information S2. This model can be used directly to make predictions for new data or used as a foundation for retraining the model using labelled data from a new application.

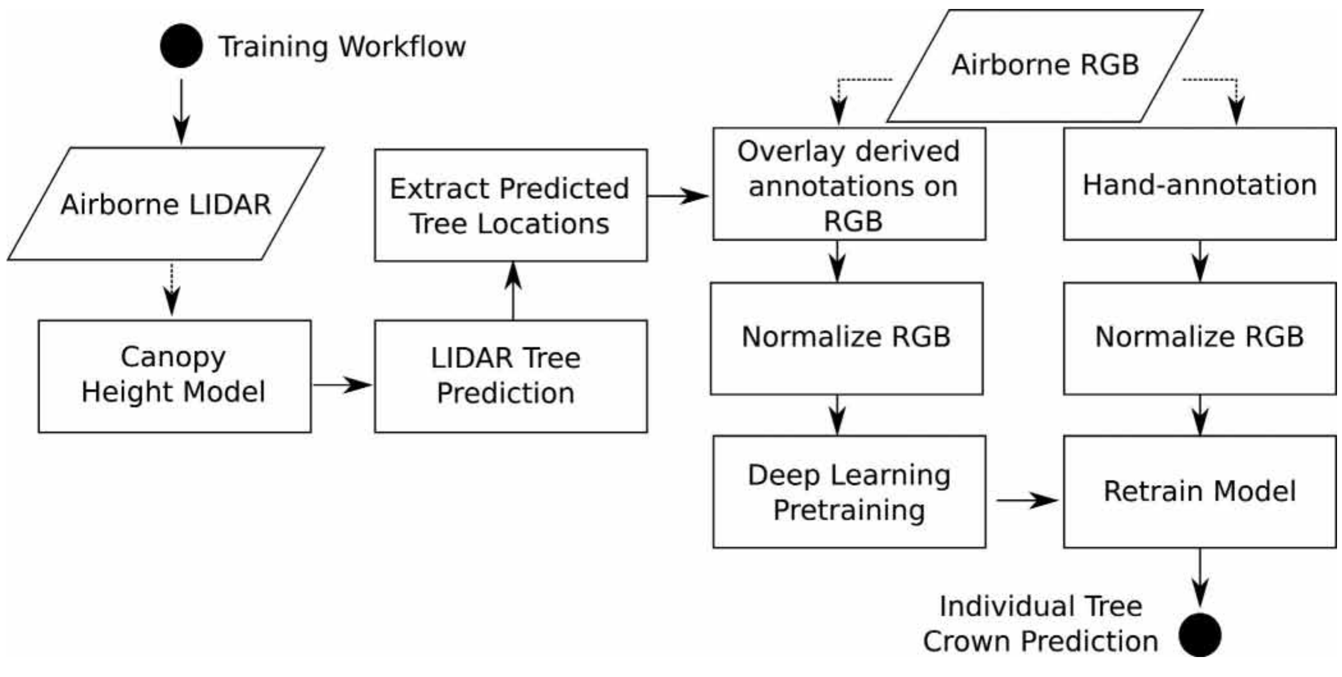

FIGURE 1 Prebuilt model training workflow. Redrawn from Weinstein, Marconi, Bohlman, et al. (2020). Parallelograms in the workflow indicate input data, rectangles indicate an algorithmic step and circles indicate the start and end of the workflow. The two sub-flows on the right-side of the figure can run in parallel and outline the pretraining and fine-tuning stages of the overall model fitting process

2.2 | Training

Tree crown delineation is a challenging task and a single model cannot realistically be expected to capture the tremendous taxonomic diversity at a global scale. This means that to perform optimal crown delineation for a particular forest requires training or fine-tuning using data from a local area. A key advantage of DeepForest's neural network structure is that users can retrain the prebuilt model to learn new tree features and image backgrounds while leveraging information from the existing model weights based on data from a diverse set of forests. Fine-tuning neural networks starting from an initial model requires less training data to produce reasonable results (Shin et al., 2016). Known as 'transfer learning', this ability is important because training deep learning models from scratch often requires tens of thousands of labelled data points for ecological tasks (Weinstein, 2018). In contrast, fine-tuning the prebuilt model with as few as 1,000 hand labelled trees can provide significant improvement and be accomplished in approximately 8–10 hr (Weinstein, Marconi, Bohlman, et al., 2020).

The standard training process starts with generating local training data by hand-labelling trees in images by placing a bounding box around each visible tree (Figure 2). This can be done using either image labelling tools (e.g. RectLabel) or GIS software (e.g. ArcGIS, QGIS) and DeepForest includes helper functions to convert common formats (XML and shapefiles) into a csv format. Annotations can be made on images of any size, but training the model requires images with fixed standard dimensions. The prebuilt model was trained on square crops of length 400 px (40 m at 0.1 m resolution), which provides a good balance between image size and providing the model landscape context for prediction. Annotation time depends on the experience of the observer and density of trees, but a 1-km tile of densely forested areas can be drawn in 8–10 hr. DeepForest includes a preprocess.split_raster function that creates a set of appropriately sized images for training using a sliding window approach. The size of these input windows is optimized for the 10 cm data used in training the prebuilt model. The upper resolution limit for tree crown delineation is currently unknown, as well as the optimal size of the input windows when performing predictions at coarser scales.

Training can be performed by fine-tuning the prebuilt model or training only using the local training data (using deepforest.train).

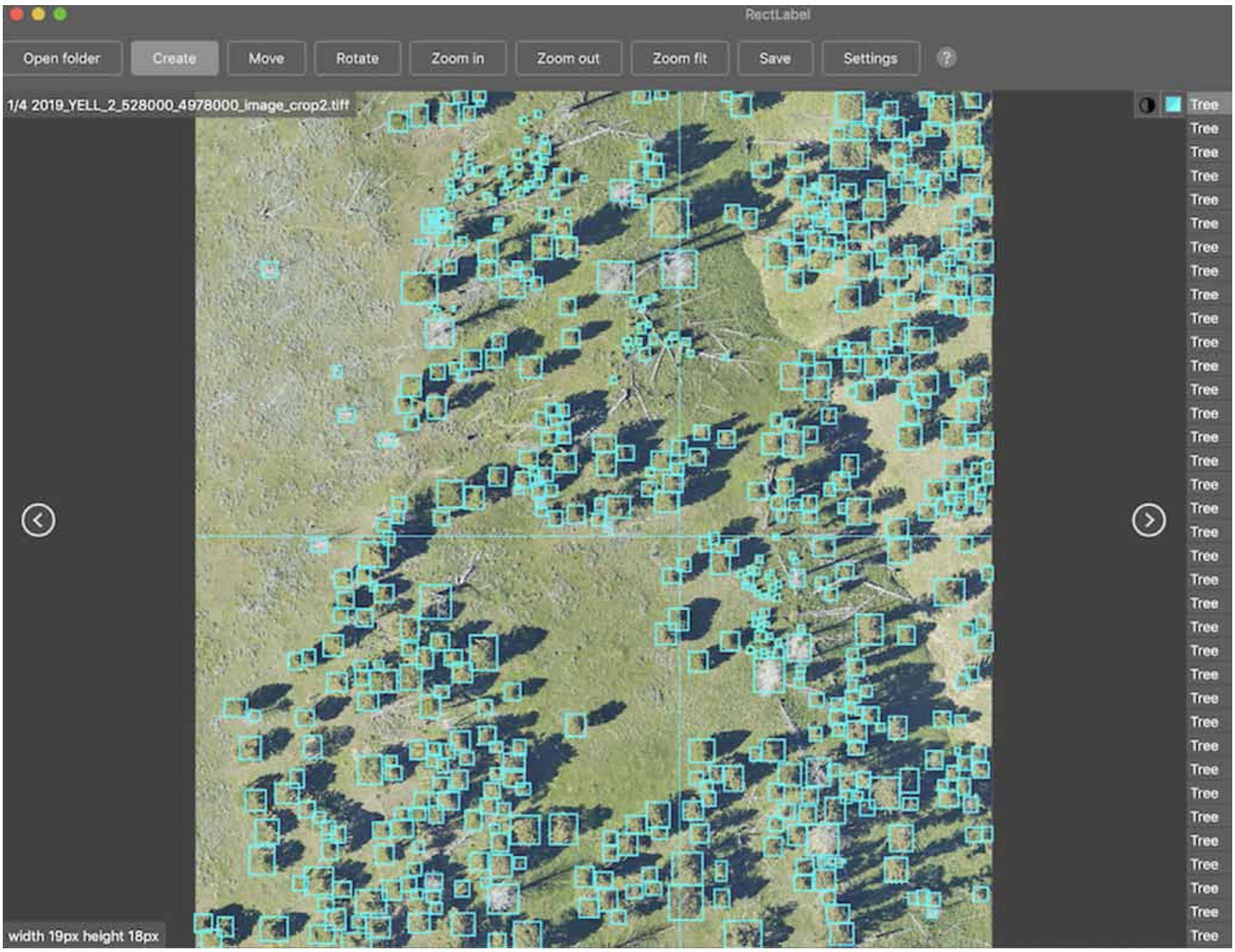

FIGURE 2 Screenshot of hand-annotated RGB image from National Ecological Observatory Network site YELL near Frog Rock, WY. For optimal training, all crowns in an image should be annotated

Training deep learning models requires a number of parameter choices such as batch size and number of epochs. For users less familiar with training deep learning models, DeepForest comes with a standard configuration file with reasonable defaults. While some parameter exploration will always be helpful, our aim is to make these innovations available even to novice users. Optional GPU support and model customization allow more experienced users to quickly develop and test larger and more complex models. Data augmentation to randomly crop and flip training images is also supported. This strategy is often useful to reduce overfitting when training on small datasets (Zoph et al., 2019) but has not been extensively tested for tree crown delineation. For additional recommendations for optimal model training see the online documentation (https://deepforest.readthedocs.io/) and Appendix S2.

2.3 | Evaluation

Deep neural networks have millions of parameters and can readily overfit, producing high scores on training data, while performing poorly on new images. This makes it essential to evaluate performance on held-out test data. To evaluate a set of annotations, users follow the same pattern as with training data: (a) Annotate one or more images of trees; (b) Cut the images into smaller windows for evaluation; and (c) Format annotations into a csv file using DeepForest's utility functions. The deepforest.evaluate_generator method can then be used to evaluate the performance of the predictions for this test data using the mean average precision (mAP). mAP combines precision and recall into a single metric measuring the area under the precision-recall curve resulting in a score ranging from 0 to 1. In our experience, mAP scores above 0.5 are usable for scientific application, but the appropriate value depends on the particular research goal and application.

2.4 | Prediction

After a model has been trained and evaluated, it can be applied to a larger collection of images to estimate the locations of trees at larger scales. High-resolution images covering wide geographic extents cannot fit into memory during prediction and would yield poor results due to the size and density of bounding boxes. DeepForest has a deepforest.predict_tile method for automating the process of splitting the tile into smaller overlapping windows, performing prediction on each of the windows and then reassembling the resulting annotations. Each bounding box annotation is returned with its xmin, ymin, xmax, ymax coordinates, and predicted probability score (the probability that the bounding box represents a tree) ranging from 0 to 1, with higher values indicating greater confidence in the prediction. To reduce overcounting among overlapping tiles, DeepForest sorts predictions by confidence scores and removes lower scoring overlapping boxes (i.e. non-max suppression).

3 | CASE STUDIES

3.1 | National Ecological Observatory Network

To evaluate model performance across a range of forest types, we used data from the National Ecological Observatory Network to predict crowns in sites across the United States. This dataset consists of 212 images containing 5,852 trees from 22 sites that is part of an upcoming tree crown benchmark data package (Weinstein, Marconi, Bohlman, et al., 2020). Training and evaluation data are separated by at least 1 km when they occur at the same site. Evaluation data were created by viewing RGB images and manually delineating tree crown boxes for all visible trees. Annotations were cross-referenced with field collected positions of tree stems (from the NEON Vegetation Structure dataset; NEON ID: DP1.10098.001) within each plot when available. Following Weinstein et al. (2019), Weinstein, Marconi, Bohlman, et al. (2020), we used precision, defined as the fraction of predicted crowns that match real trees, and recall, defined as the fraction of all evaluation trees that are correctly detected for evaluation. Following the standard evaluation for object detection in the computer vision literature (Ren, He, Girshick, & Sun, 2015), we considered predictions with Intersection over Union (IoU) scores of 0.5 as true positives. IoU, also known as the Jaccard Index, is the area of intersection between the prediction and evaluation crown, divided by the joint area of the combined prediction and evaluation crowns. We assessed the performance of the prebuilt model at all 22 NEON sites and also compared the performance to a previous version of this model (Weinstein, Marconi, Bohlman, et al., 2020) that was only trained on data from four NEON sites.

Across all sites, the average recall per image for the prebuilt model was 72% and the precision was 64%. Model performance varies across NEON sites, but most sites have both precision and recall values greater than 50% (Figure 3). The model performs similarly regardless of whether there was hand-annotated training data from the same site (Figure 3). Visual assessment of predictions across forest types reveals good overall correspondence between predicted bounding boxes and observations, with most errors resulting from insufficient overlap between observed and predicted tree crowns, rather than the model missing a tree entirely (Figure 4). The prebuilt model used by DeepForest was fit to data from 22 NEON sites and outperforms the previous four-site model (Weinstein, Marconi, Bohlman, et al., 2020) at 19 of 22 sites for recall and 16 of 22 sites for precision, demonstrating that increasing the diversity and amount of training data has improved the performance of the model. These results demonstrate that the prebuilt model can make reasonable predictions in forests ranging from deciduous forests of the Northeast, to southern pinelands, to coniferous forests of the mountain west.

The site with the worst performance is Onaqui, Utah (ONAQ), which is a desert scrub site with a different vegetation structure from any of the training data. The site is almost treeless and includes trees with short and gnarled stature. This highlights the importance of using local training data to reduce uncertainty when working with data that are not well represented in the training data for the prebuilt

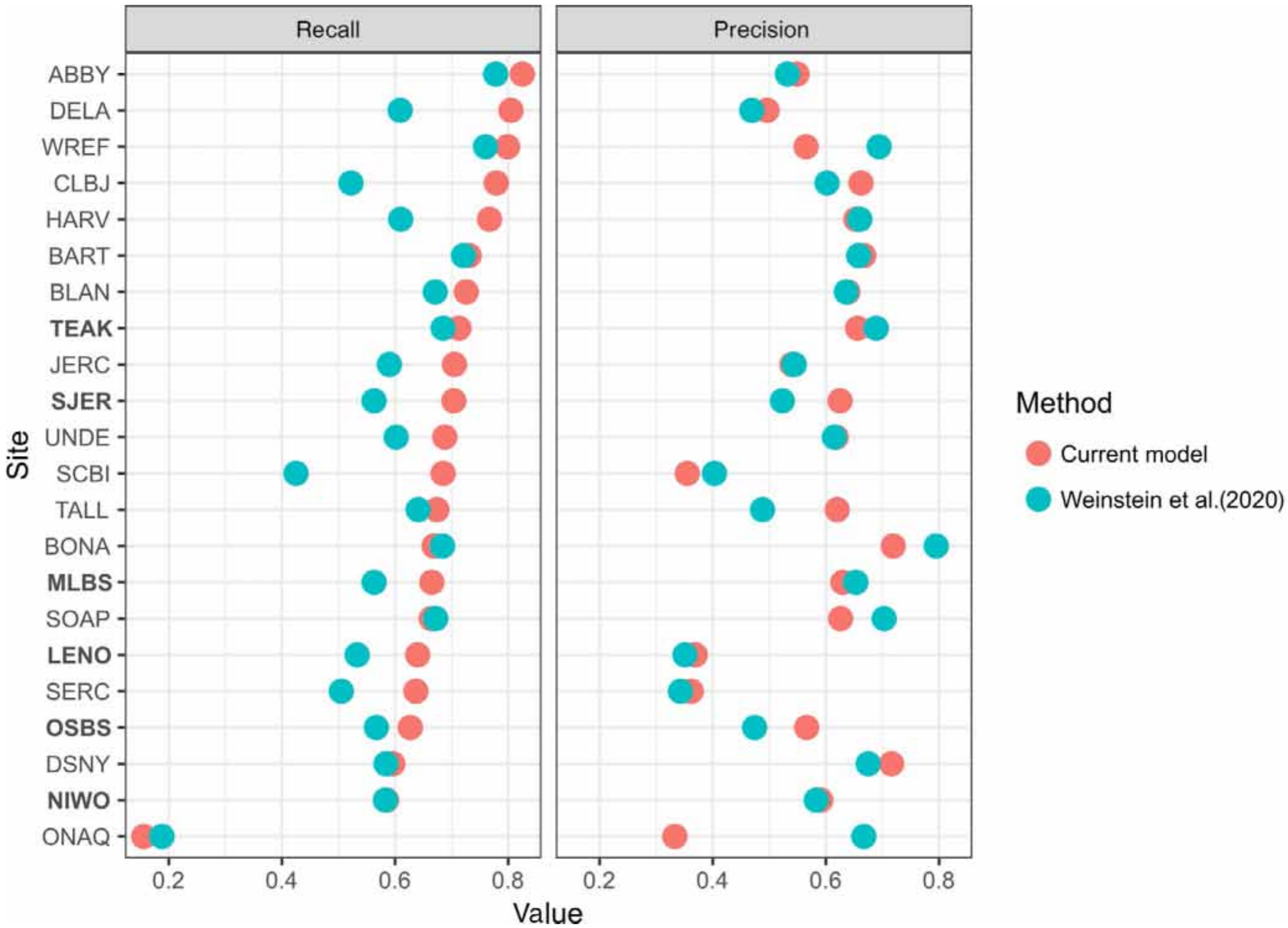

FIGURE 3 Precision and recall scores for hand-labelled evaluation images from the National Ecological Observatory Network (current prebuilt model in red, Weinstein, Marconi, Bohlman, et al., 2020 in blue). Sites in bold had hand-labelled data included in training the current prebuilt model. See Supporting Information S1 for site abbreviations

model. In these contexts, the value of the prebuilt model is that it reduces the needed training sizes when applied to new conditions. This has the potential to support training with small amounts of data for applications to a wide array of questions surrounding tree health and ecology. For example, training a model specific to bare trees could allow studies of broad-scale pest outbreaks or timing of deciduous phenology. Initial tests at the Soaproot Saddle, CA site ('SOAP' in Figure 4, third row) show the prebuilt model can detect standing dead trees when visible. Adding additional training data could allow broad-scale analysis of tree health when comparing images across time.

3.2 | French Guiana tropical forest

The DeepForest prebuilt model was trained on data from the United States that was collected using fixed-winged aircraft at 10 cm resolution and provided as 1 km^2 orthomosaics. Therefore, two key questions are: (a) does this model generalize to images collected in new locations or using different acquisition hardware and (b) how useful are the (re)training features of the software for improving performance in novel contexts? It is also important to understand how the DeepForest RGB model compares to LiDAR-based models from recently published work.

To address these questions, we used data from a recently published competition comparing LIDAR tree segmentation algorithms using remote sensing from French Guiana (Aubry-Kentz et al., 2019). In the original competition, each team was sent unlabeled data to predict and the evaluation data was kept private. This process was repeatled for this paper, with the third author (Aubry-Kientz) running evaluation scores for the DeepForest predictions made by the corresponding author (Weinstein). Predictions were run on a Mac laptop with a 3.1 GHz Intel Core i5 processor. Predictions from each algorithm were compared to hand-delineated evaluation crowns based on field observation and manual comparison with RGB and LiDAR data. Validation crowns were delineated as polygons, rather than the rectangular bounding boxes generated by DeepForest. This case study also provides information on whether DeepForest's approach of predicting rectangular bounding boxes leads to lower prediction accuracy than methods producing polygons. Algorithm recall was scored based on the proportion of labelled trees predicted with IoU scores of >0.5. Precision was not calculated because not all crowns in the test imagery were delineated (see Figure 5).

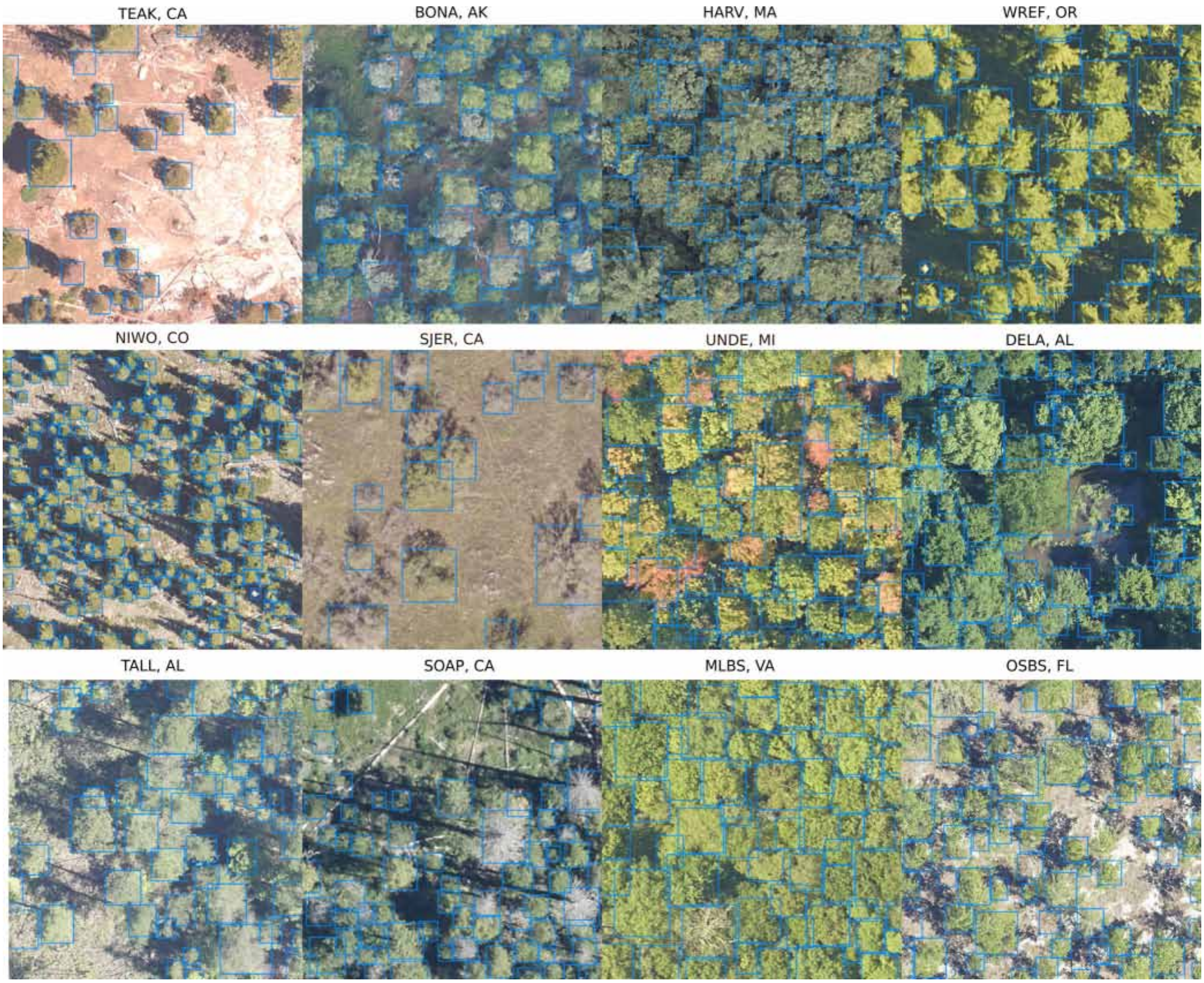

FIGURE 4 Panel of tree predictions from a broad range of evaluation images in the National Ecological Observatory Network with predicted tree crown boxes in blue. Each image is labelled with the National Ecological Observatory Network site abbreviation and state. See Supporting Information S1 for site abbreviations

FIGURE 5 RGB images collected over a tropical forest in French Guiana and example of manually segmented crowns used to evaluate the segmentation

We used DeepForest to detect tree crowns using three approaches (a) the prebuilt DeepForest model with no local training; (b) a model fit solely to 5,018 local hand-annotated crowns (annotated by BW using only the RGB data on tiles separate from the evaluation data) and (c) the prebuilt model fine-tuned using the local annotations. RGB tiles were divided into 800 px windows for model training and evaluation. The default patch size of 400 px was increased to 800 px to minimize the edge effect of overlapping crowns. Models 2 and 3 were trained for seven epochs with a runtime of approximately 11 min/CPU on a laptop, which demonstrates that while advanced GPU hardware is convenient for training large datasets, fine-tuning and training on small datasets can be done locally on CPU.

The prebuilt model performed well on this novel data with a recall of 0.64, close to the 0.71 recall for the best performing LIDAR based algorithm from Aubry-Kientz et al. (2019). Training only on the 5,018 local annotations resulted in a poorer recall of 0.35. Retraining the prebuilt model with the local annotations produced the best results with a recall of 0.78, slightly better than the highest performing LiDAR algorithm from Aubry-Kientz et al. (2019). This analysis is not sufficient to draw general conclusions about RGB versus LiDAR-based methods, but these results do suggest that DeepForest is competitive with state-of-the-art LiDAR-based approaches. Overall, the case study demonstrates the utility of DeepForest both using the prebuilt model and using local retraining to improve crown delineation based on local conditions.

3.3 | Portland street trees

DeepForest's use of widely available RGB data provides the potential for it to be used across very large spatial extents. Scaling up is challenging because algorithms need to handle large ranges of habitat types and because the resolution of the data available over large areas is typically coarser. To explore how DeepForest performs using coarser resolution data in unique habitats, we applied both the prebuilt model and a retrained model to crown delineation of street trees in an urban environment. The locations of urban trees are important for ecological, sociological and public infrastructure applications. In addition, the urban environment is very different from the natural environments on which the prebuilt model was trained. The image data from the Oregon Statewide Imagery Program is also coarser at 0.3 m spatial resolution (1 ft), a resolution that is widely available as part of the National Agriculture Imagery Program (NAIP—https://www.fsa.usda.gov/programs-and-services/aerial-photography/imagery-programs/naip-imagery/).

We used imagery from the Portland metro area that overlapped with the Portland Street Trees dataset (http://gis-pdx.opendata.arcgis.com/datasets/street-trees). The street trees dataset contains geospatial information for the majority of trees accessible from public roads in the metro area. Not all trees in an image are labelled, since many trees occur on private property and are not mapped. We divided the RGB imagery into geographically distinct training and test datasets and used the street trees dataset to guide hand-annotation of a small number of tree crowns (n = 1,033). Annotation by hand took approximately 3 hr and covered a small geographic area of mixed urban development, empty lots and ballfields (Figure 6). The street trees data were collected prior to the RGB images and were cleaned to remove trees which were cut down or were obvious errors (e.g. trees located in the middle of buildings). To evaluate the street tree case study, we used field collected location of the tree stems to measure tree recall and the rate of undersegmentation. Recall was defined as the proportion of street tree locations that were contained within a predicted tree bounding box. Undersegmentation rate was defined as the proportion of predicted boxes that matched more than one street tree. Minimizing undersegmentation is challenging because trees growing close together can appear to be a single tree from above and is therefore best evaluated against ground collected data.

We found that evaluating and retraining on data with coarser resolution than the prebuilt model required careful choosing of the size of the focal view. The prebuilt model was originally trained on a 40 m focal view (400 px windows with 0.1 m data). Data exploration on the coarser data source showed that larger focal views of 60–120 m performed better than maintaining the original 40 m view, and 60 m was chosen for this analysis. In general, we expect that the focal view size should increase with coarser resolution data, but this remains an area of further exploration.

As with the tropical forest case study, we found that the prebuilt model performed reasonably well (recall = 0.55; undersegmentation = 0.25) and retraining with a small amount of local

FIGURE 6 Predictions made on a tropical forest in French Guiana using the prebuilt model retrained with local annotations. Each individual tree is labelled with a blue bounding box

FIGURE 7 Predictions for the Portland street tree case study. Bounding box predictions from the prebuilt model are in orange. Bounding box predictions from the retrained model using local data are in blue. Street tree locations are marked in purple

training data significantly improved algorithm performance with an increase in recall and decrease in undersegmentation (recall = 0.72; undersegmentation = 0.17; Figure 7). Visual inspection shows that many of the errors in using the retrained model are for small trees difficult to resolve in the imagery, or tree types not present in the limited training data (e.g. ornamental trees with a deep purple hue).

4 | CONCLUSION

DeepForest provides an open source software package for: (a) delineating tree crowns in RGB imagery, (b) evaluating the performance of that crown delineation using hand labelled evaluation data and (c) training new models and fine-tuning of the included prebuilt model to support prediction tailored to specific forest types. The inclusion of a prebuilt model allows users to benefit from the strengths of deep learning without needing to deal with many of the challenges. Given the enormous diversity of tree appearance at a global scale, defining a single unified model for tree crown delineation is challenging. To address this, DeepForest provides an explicit retraining method to improve performance for specific use cases. This allows the user to decide what level of accuracy is required for the target question, and then annotate local data and retrain the model to produce predictions with sufficient accuracy for their use case. We recommend defining a clear evaluation dataset, setting a threshold for desired performance before training and using evaluation data that is geographically separate from the training data to ensure that the prediction threshold holds outside of the training region.

The minimal spatial resolution for accurate tree prediction using this software remains unknown and may ultimately relate to the desired ecological or management question. Analysis of the NEON data shows that individual tree segmentation is achievable at 10 cm. The Portland street trees example shows that 30 cm data (which is publicly available for many states and counties) provides reasonable delineations. However, the accuracy will not be as high as with higher resolution data, and further analysis at this resolution is necessary. One meter resolution imagery is increasingly available at near continental scales (e.g. NAIP 1 m imagery which provides nearly complete coverage of the United States; https://www.fsa.usda.gov/programs-and-services/aerial-photography/imagery-programs/naip-imagery/). It is unlikely that these data will be effective at distinguishing small individual trees, but it may be useful in identifying large trees or clusters of trees in sparse landscapes.

To support the broad application of predictions from DeepForest, these predictions can be easily exported for use in further analysis and combination with other sensor products for forest research. Individual tree crown delineation is often the first step in key remote sensing analyses of forested landscapes, including biomass estimation (Kamoske, Dahlin, Stark, & Serbin, 2019), species classification (Maschler, Atzberger, & Immitzer, 2018), and leaf-trait analysis (Marconi, Graves, Weinstein, Bohlman, & White, 2020). DeepForest both ingests and outputs crowns in an easily accessible, standardized annotation format, and will facilitate further improvements in the prebuilt model based on community contributions.

ACKNOWLEDGEMENTS

We thank Brady Callahan for his assistance in accessing Oregon imagery data and the Portland Metro for their work collecting the Portland Street Trees dataset. This research was supported by the Gordon and Betty Moore Foundation's Data-Driven Discovery Initiative through grant GBMF4563 to E.P. White and by the National Science Foundation through grant 1926542 to E.P. White, S.A. Bohlman, A. Zare, D.Z. Wang and A. Singh. We thank Alina Zare, Aditya Singh and Dylan Sinnickson for feedback during model and software development and comments on the manuscript. The authors declare no conflicts of interest.

AUTHORS' CONTRIBUTIONS

B.G.W., S.M., E.P.W. conceived of the project, designed the package and wrote the manuscript. M.A.-K., G.V. and H.S. performed analysis, provided package improvements and edited the manuscript.

PEER REVIEW

The peer review history for this article is available at https://publons.com/publon/10.1111/2041-210X.13472.

DATA AVAILABILITY STATEMENT

DeepForest source code is available on GitHub (https://github.com/weecology/DeepForest) and archived on Zenodo: https://doi.org/10.5281/zenodo.3906928 (Weinstein, Marconi, & White, 2020). The code for the case studies is available in a separate repo (https://github.com/weecology/DeepForest_demos). The inbuilt development version of the NEONTreeEvaluation benchmark is available online (https://github.com/weecology/NeonTreeEvaluation) and will continue to be updated as more images are annotated. The data for the Street Trees example is archived on zenodo: https://zenodo.org/record/4047083#.X3IEhpNKhhE.

ORCID

Ben G. Weinstein https://orcid.org/0000-0002-2176-7935
Ethan P. White https://orcid.org/0000-0001-6728-7745

REFERENCES

Aubry-Kientz, M., Dutrieux, R., Ferraz, A., Saatchi, S., Hamraz, H., Williams, J., … Vincent, G. (2019). A comparative assessment of the performance of individual tree crowns delineation algorithms from ALS data in tropical forests. *Remote Sensing*, *11*, 1086. https://doi.org/10.3390/rs11091086

Coomes, D. A., Dalponte, M., Jucker, T., Asner, G. P., Banin, L. F., Burslem, D. F. R. P., … Qie, L. (2017). 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. https://doi.org/10.1016/j.rse.2017.03.017

Hamraz, H., Contreras, M. A., & Zhang, J. (2016). A robust approach for tree segmentation in deciduous forests using small-footprint airborne LiDAR data. *International Journal of Applied Earth Observation and Geoinformation*, *52*, 532–541. https://doi.org/10.1016/j.jag.2016.07.006

Kamoske, A. G., Dahlin, K. M., Stark, S. C., & Serbin, S. P. (2019). Leaf area density from airborne LiDAR: Comparing sensors and resolutions in a temperate broadleaf forest ecosystem. *Forest Ecology and Management*, *433*, 364–375. https://doi.org/10.1016/j.foreco.2018.11.017

Marconi, S., Graves, S., Weinstein, B., Bohlman, S., & White, E. (2020). Rethinking the fundamental unit of ecological remote sensing: Estimating individual level plant traits at scale. *bioRxiv* 556472. https://doi.org/10.1101/556472

Maschler, J., Atzberger, C., & Immitzer, M. (2018). Individual tree crown segmentation and classification of 13 tree species using airborne hyperspectral data. *Remote Sensing*, *10*, 1218. https://doi.org/10.3390/rs10081218

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. *Nips*, *91–99*, https://doi.org/10.1109/TPAMI.2016.2577031

Shin, H., Roth, H. R., Gao, M., Lu, L., Member, S., Xu, Z., … Summers, R. M. (2016). Deep convolutional neural networks for computer-aided detection: CNN architectures. *Dataset Characteristics and Transfer Learning*, *35*, 1285–1298. https://doi.org/10.1109/TMI.2016.2528162

Silva, C. A., Hudak, A. T., Vierling, L. A., Loudermilk, E. L., O'Brien, J. J., Hiers, J. K., … Khosravipour, A. (2016). Imputation of individual longleaf pine (*Pinus palustris* Mill.) tree attributes from field and LiDAR data. *Canadian Journal of Remote Sensing*, *42*(5), 554–573. https://doi.org/10.1080/07038992.2016.1196582

Weinstein, B. G. (2018). A computer vision for animal ecology. *Journal of Animal Ecology*, *87*, 533–545. https://doi.org/10.1111/1365-2656.12780

Weinstein, B. G., Marconi, S., Bohlman, S., Zare, A., & White, E. (2019). Individual tree-crown detection in RGB imagery using semi-supervised deep learning neural networks. *Remote Sensing*, *11*, 1309. https://doi.org/10.3390/rs11111309

Weinstein, B. G., Marconi, S., Bohlman, S. A., Zare, A., & White, E. P. (2020). Cross-site learning in deep learning RGB tree crown detection. *Ecological Informatics*, *56*, 101061. https://doi.org/10.1016/j.ecoinf.2020.101061

Weinstein, B. G., Marconi, S., & White, E. P. (2020). Python package for tree crown detection in airborne RGB imagery (Version v0.3.0). *Zenodo*, https://doi.org/10.5281/zenodo.3906928

Zhu, X. X., Tuia, D., Mou, L., Xia, G.-S., Zhang, L., Xu, F., & Fraundorfer, F. (2017). Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*, *5*, 8–36. https://doi.org/10.1109/MGRS.2017.2762307

Zoph, B., Cubuk, E. D., Ghiasi, G., Lin, T. Y., Shlens, J., & Le, Q. V. (2019). Learning data augmentation strategies for object detection. arXiv preprint arXiv:1906.11172.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section.

How to cite this article: Weinstein BG, Marconi S, Aubry-Kientz M, Vincent G, Senyondo H, White EP. DeepForest: A Python package for RGB deep learning tree crown delineation. *Methods Ecol Evol*. 2020;11:1743–1751. https://doi.org/10.1111/2041-210X.13472