

Tropical Forest Carbon Stock Estimation using RGB Drone Imagery

Victoria Barenne

ETH Zurich

Department of Mathematics

vbarenne@student.ethz.ch

Jan Philipp Bohl

ETH Zurich

Department of Mathematics

jabohl@student.ethz.ch

Dimitrios Dekas

ETH Zurich

Department of Computer Science

ddekas@student.ethz.ch

Tim Engelmann

ETH Zurich

Department of Mechanical and Process Engineering

tengelmann@student.ethz.ch

Abstract

Automating the estimation of carbon stock for forestry sites is an important yet challenging task due to the inherent diversity of the problem. Recently, the advancements in machine learning algorithms, and especially deep learning techniques, gave rise to previously unexplored solutions. However, in order to tackle this problem using deep learning, it is necessary to combine efforts and collaborate. To that end, the benchmark forestry dataset *Reforestree* was created using a collection of diligently hand collected measurements and drone images [9]. In the current work, we further explore *Reforestree* and strive to bring light to its shortcomings and limitations. We propose a collection of approaches for estimating carbon stock from images, while tackling the GPS noise, matching discrepancies and outlying samples found in the underlying dataset. Our methods significantly reduces the need for pre-processing procedures that inevitably produce additional noise in the final dataset. Furthermore, they are able to account for the GPS noise and smooth out the carbon distribution. Finally, a comparison between the pipeline found in the original paper and the ones developed in this work is presented. The trained models are unable to make meaningful predictions on unknown data and fail on capturing the underlying carbon distribution.

1. Introduction

In recent years, climate change and its consequences have gathered a significant amount of attention. Governments and institutions around the world invest considerable funds in research and development, seeking new ways to reduce their carbon footprint. In addition, carbon offsets motivate entities, such as companies, to invest in develop-

ing countries and their forestry sites. However, keeping track of the carbon stock present in a particular site is costly due to its demand for labor and technological tools. Having a firm knowledge of the carbon stored in forest sites can prove of substantial importance. Reducing monitoring, verification, and reporting cost can have a significant impact on policy making, both for governments and corporations. Therefore, carbon stock estimation is an active research topic and has attracted increasing attention over the last few years [6] [12].

Methods addressing carbon stock estimation can be roughly classified into manual and imagery based. In order to calculate the carbon stock of a given site, manual methods depend on field measurements that are collected by experts for a sample of trees. The aboveground biomass (AGB) can be calculated, using a series of metrics for every tree and by taking advantage of allometric equations. Then, the total carbon stock of a given forestry site can be easily approximated, through standardized calculations [11]. However, these approaches are known to usually overestimate the actual ground truth since the use of labor can lead to human mistakes and subjective judgments when it comes to measurements.

On the other hand, satellite-based methods utilize deep learning techniques and lidar as tools to produce vast estimations for the carbon stock of entire forests [13] [2]. The main advantage that these new approaches offer is a reduction in resource consumption, both time and cost-wise. One of the pivotal limitations of satellite-based carbon calculation methods is the considerable overestimation of the ground truth amount. Additionally, satellite-based approaches are only suitable for carbon estimations of large-scale forestry sites. Therefore, these methods do not allow for the incorporation of smaller investors in the carbon offsetting market and slow down its democratization.

In this work, we explore a recently published dataset named ReforesTree [9]. We are interested in better understanding the process that created the dataset and ensuring the truthful delivery of its claims. We produce visual representations for the carbon distributions that govern the forestry sites and question the pre-processing pipeline proposed by the original work. By doing so, we are able to detect the main issues present in the dataset, some of which are the GPS noise, wrong matching, and outlying carbon values. In an attempt to account for these problematic attributes, we propose a variety of new approaches. We introduce the concepts of patching, deriving a Gaussian distribution for the carbon stock of each site and using a tree density based label scheme. Finally, for each approach, we train a ResNet-18 architecture and compare our results.

The following sections can be described as following. Sec 2 introduces the original ReforesTree publication and relevant works. Sec 3 gives an overview for the analysis of the dataset and lands the foundation for the proposed approaches. Sec 4 discusses the methods developed in order to combat the flaws detected in the dataset. Sec 5 presents the experimental results of the proposed methods. Sec 6 summarizes our findings and utilizes them to draw final conclusions.

2. Related Work

2.1. Dataset

The present work is based on the ReforesTree dataset published in [9]. It consists of a set of tree crown images and their associated measurements from six tropical agroforestry sites located in Ecuador, as shown in Fig 1. For every site present in the dataset, aerial images were captured using a drone equipped with an RGB camera. Along with the imagery data, field measurements were produced for a number of trees on each site. These include the tree GPS location, diameter at-breast-height (DBH) and species. The data acquisition was carried out by WWF Switzerland and the raw data can be found in the ReforesTree repository [10]. The final ReforesTree dataset is produced by deploying a pre-processing pipeline on the collected images and measurements.

2.2. Original Pipeline

The original pipeline consists of two main sub-processes. The first one makes use of the drone images to extract individual tree crown images, more specifically it extracts bounding boxes and their coordinates. The second subprocess calculates the AGB values of each individual tree, based on their DBH and species, and matches them to their corresponding image using a greedy optimal transport algorithm. In this section, we will dive deeper into the details of the tree crown detection and matching algorithms. We will



Figure 1. The six forestry sites located in Ecuador that can be found in the dataset

also explore how the final dataset is processed and used to train models.

Tree Crown Detection The first step in the original pipeline is to extract individual tree crowns from the RGB drone images. To do so, an already existing tree crown detection model DeepForest [14] was fine-tuned. This deep-learning architecture takes a 4000x4000 RGB image as input and outputs bounding boxes for all detected tree crowns. Originally, the DeepForest model was developed using drone images of forests located in North America, whose vegetation differs greatly from that of the tropical forests in Ecuador. Thus, to improve performance across the Ecuador sites, the network was fine-tuned using 324 hand-labeled tree crown bounding boxes. The detected bounding boxes were then padded with white pixels or center cropped to size 800x800 pixels.

Matching Algorithm To match the detected tree crowns to the recorded field data, a special implementation of optimal transport was developed in the following publication [1]. This implementation uses a regularized optimal transport algorithm with greedy matching while also leveraging species labels.

The optimal transport algorithm used for this approach is the Sinkhorn algorithm: it determines the optimal cost of moving one distribution to another, adding a regularization term to avoid overfitting [8]. The result is a transport plan detailing for each bounding box and field data point the probability that they should be matched. To address the surplus of tree crowns compared to field data, a greedy matching strategy is applied to the transport plan. Bounding boxes are matched to field data, by finding the most likely match in the entire transport plan. The corresponding pair of bounding box and field data point are then removed from

the transport plan and the next most likely match is used until all bounding boxes are attributed to field measurements. This approach ensures that the matching is one-to-one and that in the case of an unequal number of field data points and bounding boxes, no data is matched to more than one bounding box or vice versa.

To leverage the species labels, a classifier is trained on hand-labeled data to distinguish banana tree crowns from non-banana tree crowns as the former are easily recognizable. All tree crowns are then classified using the resulting model. These labels can then be used to separately match banana and non-banana tree crowns and field measurements.

Prediction To evaluate the quality of the final Reforestree dataset, the images were resized to 224x224 pixels and then split into 70% training and 30% testing randomly across all sites. A pre-trained ResNet18 convolutional neural network [5], with the last layer modified to change it from a classification model to a regression model, was then trained using mean-square-error loss.

3. Analysis of Dataset

The Reforestree dataset provides tree crown images and AGB labels as a machine learning ready dataset. In the current section, we describe several shortcomings with this data: the GPS locations of field measurements are noisy, this data also has several inconsistencies, the tree crown detection fails for dense forests and too many trees were detected because of excess image data. These errors affect the matching of tree crown images and field data leading to questionable labels in the final dataset.

3.1. Boundary

The raw dataset provides a report on the data collection process. This report explains that the provided drone images cover more than the site considered for estimating carbon. Only parts of these images correspond to the agroforestry sites where field data was collected. However, the authors considered the entire site for tree detection and subsequently tried to match the resulting bounding boxes to the field measurements. To account for this, the excess image data needs to be removed. We considered two methods to do so: using the shape files provided in the dataset and creating a boundary from the field data. The first method would be the natural choice, however, the boundaries provided are of poor quality for some sites as it can be seen in Fig A.3.

We thus propose creating a boundary based on the field data, for which we considered two algorithms: using their convex hull or the more complex alpha shape. The former will extend beyond the actual site because of the GPS error

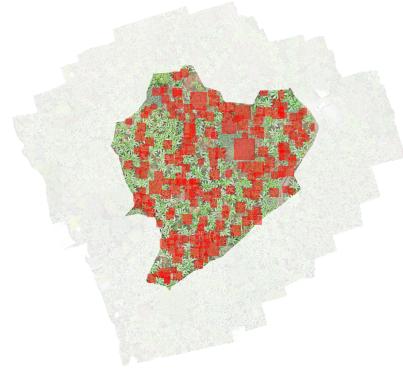


Figure 2. Bounding boxes and boundary for Flora Plus site

and the convexity. This issue can be reduced by using alpha shapes which allow for a tighter bound around the data points. Alpha shapes are a generalization of convex hulls first described in the following paper [3]. They allow for a tighter bound around the data points.

As shown in Fig 2, when the images are restricted to an alpha shape around the field data, a part of the drone image is removed. Detecting trees on the entire image would thus lead to a significant overestimation of the number of bounding boxes.

3.2. Tree crown detection

To obtain tree crown images from the drone images, the authors used the DeepForest [14] pipeline. The model struggles to detect all trees in the denser tropical forests present in this dataset. When the excess image data is removed as described in the previous subsection, we are left with few detected tree crowns (1564) compared to field data points (4663), which causes issues in the matching. Fig 2 shows that inside the alpha shape boundary, many trees are not detected. This indicates that in the current state, the DeepForest pipeline is inadequate for tree detection in dense tropical forests, which also means that the pipeline proposed by the authors will have difficulty correctly estimating the carbon stored in the agroforestry sites.

3.3. GPS Noise

The noise in the GPS position of tree measurements is considerable. This can already be seen when plotting these positions over the RGB Image data, as in Fig 3. The locations do not match with the actual trees and do not respect the patterns seen on the more structured sites (trees planted in rows). This GPS noise makes it hard to correctly assign the carbon values measured on site to the drone images, resulting in noisy labels. To quantify the GPS error we aim

to match the field data distribution and the actual tree positions. An optimal match represents a lower bound on the GPS error. For our analysis, we focus on the most structured site *Leonor Aspiazu* and only consider *Musacea* (banana) trees, as those are the easiest to identify. We hand label all of those trees by drawing a bounding box around the tree crown. Within the adapted site boundary we arrive at 319 labels. However, we only have recorded field data for 210 *Musacea* trees. It is hard to judge why these numbers differ, potential reasons are: a) We mistake one larger tree as two smaller ones, b) we include some really small trees that might have been discarded during field measurements, c) we confound some cacao trees with banana trees and/or d) the field data is inaccurate.

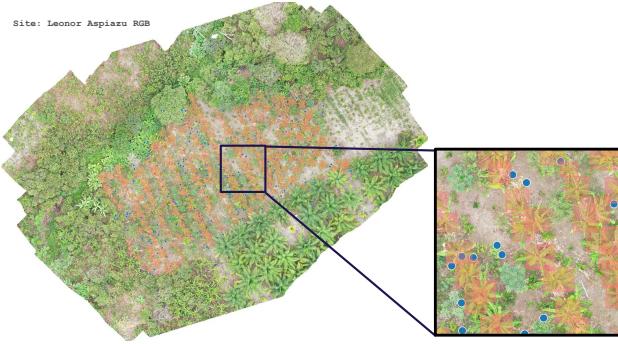


Figure 3. *Musacea* trees within field data (blue) and hand annotated bounding boxes (red). The field data points do not coincide with actual trees.

To continue with our analysis and account for error a) we decide to perform K-Means Clustering on the center coordinates of our bounding boxes and arrive at 210 new labels. Next, we match those with the field data coordinates by applying an optimal transport algorithm with euclidean distance and no regularization [4]. The distributions and corresponding matches are shown in A.1. We report a Wasserstein metric of $W_1 = 432.89m$. We then pair each field data point to the target label with the highest probability and calculate the distance between the two. We arrive at a GPS error estimate of $433 \pm 308px$. Using the ground sampling distance provided within the dataset documentation, this corresponds to $4.59 \pm 3.27m$. We also fit a 2D Gaussian on the matched vectors, which is given in A.2 and will be reused in Sec 4.2. Given that this error represents a lower bound only, it is questionable if the dataset remains usable.

3.4. Matching

The errors present in the dataset, in particular the GPS noise and the problems in tree crown detection, make it difficult to meaningfully match tree crowns to field measurements. The use of bounding boxes detected on the entire images results in faulty matching: the optimal transport

algorithm tries to stretch the field data points outwards to cover the entire image. As detailed above, we know that this should not be the case. Even when only using bounding boxes in the vicinity of the measurements, the surplus of field data points compared to bounding boxes ensures that many field measurements can be matched to a single tree crown. This matching will be incorrect in many cases, due to the inaccuracy in the GPS location of the field data. This is particularly problematic when some of the outlier field measurements with high carbon values are matched to tree crown images of average trees which should have significantly lower carbon values.

3.5. Outlier Trees

The existence of outliers in the dataset is a known problem and is also mentioned in the original work [9]. As explained by the authors, there are certain trees with a DBH value that is unexpectedly large for their respective species. In order to account for that discrepancy they proposed a simple imputation scheme. They impose a threshold for the DBH of a tree equal to 50cm and decide to clip any larger diameter. They set the clipping value equal to the maximum DBH of same-species trees that were planted in the same year as the outlying tree.

It is worth mentioning that, there are trees with high DBH values that are expected to showcase that behavior, and therefore, the authors allow for their diameter to remain untouched. For instance, timber trees display a higher DBH when compared to musacea and cacao trees but do not necessarily constitute outliers. In our own exploration, and by carefully visualizing the carbon distribution of the dataset, we noticed another unexpected discrepancy, this time, considering the carbon values of trees. There exist certain trees that showcase great dissimilarities in the carbon values attributed to them when compared to the mean of their respective species.

3.6. Size and Non-Homogeneity

The original ReforesTree dataset contains 4663 detected tree crowns across the six sites. Once we filter out the out-of-bounds crowns, this number is reduced to 1564. This is very little data to train large networks such as ResNet18 [5], which consists of around 11 million trainable parameters, and could lead to overfitting. Furthermore, we also notice that the six sites are highly inhomogeneous, with some sites having neat and spaced-out rows of trees and other sites containing clusters of trees that are hard to differentiate. Overall, the amount of data is small, whether it be in the number of sites or individual trees. As such, we decided on methods that would avoid any loss of the data available to us.

4. Methods

To address the problems with the previous work and the dataset, we propose several strategies. First, instead of detecting individual tree crowns and predicting their associated carbon values, we split the images into patches and predict the stored carbon for each. This removes the error introduced by the DeepForest tree crown detection and bypasses the need for tree-level matching. Second, to further reduce the dependence on the GPS error, we assume that the carbon is not stored exactly at the provided location, but follows a normal distribution around it. A sketch of this approach is shown in Fig 4. Third, we try to tackle the problem of inhomogeneous data and large carbon values. We assume the same carbon value for each tree and therefore arrive at a smoother carbon distribution.

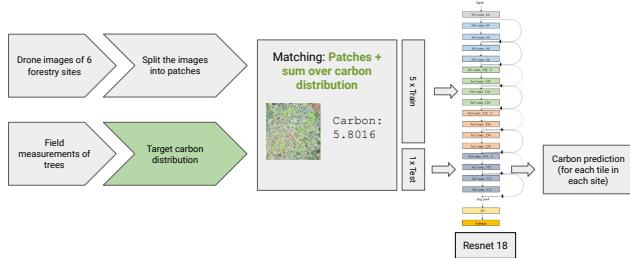


Figure 4. Overview of our proposed approach, using patching and assuming a gaussian carbon distribution.

4.1. Patching

Our first approach consists in splitting each site image into smaller patches of equal size. Each patch is then associated to its carbon value, which is just the sum of the carbon of all trees within the patch. This allows our models to see all trees present on the sites, which is crucial given the limited data. To ensure that this method produces patches of equal size, we pad the images and carbon distributions using white pixels and zero carbon pixels respectively so that their dimensions are divisible by the patch size. The padding is done in the bottom right corner for both so that the coordinates remain valid.

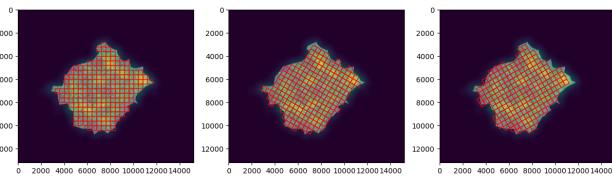


Figure 5. Example of patching using rotation angles $0^\circ, 30^\circ, 60^\circ$ on the *Nestor Macias* site.

To allow for data augmentation, we also decide to in-

clude rotations of the site images before patching, as in Fig 5. The choice of rotation angles and patch size then determines how much data is generated, and they can be considered as hyperparameters.

4.2. Gaussian

Instead of assuming the entire carbon value of a tree to be located exactly at the recorded GPS location, we assume that it is normally distributed around that point. This uncertainty represents the error in the GPS location. The mean and variance of the gaussian are computed by matching field data recordings to hand-labeled bounding boxes of banana trees in a single site, as described in Sec 3.3. We reuse the same gaussian parameters for all sites and thus implicitly assume that the GPS error is similar across sites, warranted by the fact that they all lie in Ecuador. Following this approach, we arrive at a carbon distribution for each site. To compute the carbon label for a specific patch, one can simply sum over its carbon distribution. Fig 6a shows the carbon distribution for a site, where its trees have similar carbon values. However, some sites include trees with outlying high carbon values, as described in Sec 3.5. Fig 6b shows that the carbon distribution of such a site is still highly imbalanced. It is also worth mentioning that due to the nature of the gaussian distribution, some of the carbon will propagate out of bounds. The quantity of that carbon loss highly depends on the chosen variance and site boundary. However, this effect only matters if one is interested in predicting the overall carbon for a site rather than for individual patches.

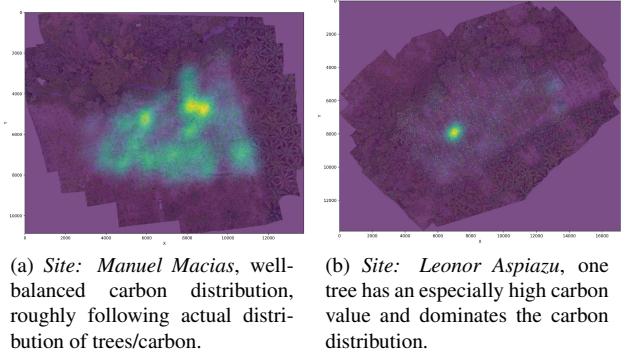


Figure 6. Gaussian carbon distributions of two sites.

4.3. Tree Density

As mentioned in Sec 3.5, a number of trees exhibit outlying carbon values. To alleviate this issue, we propose an alternative value for the labels used during training. Specifically, we modify the previously discussed pipeline to derive a tree density based estimation scheme. For this approach we firstly deploy the Gaussian distribution to extract an estimate for the tree density in a particular patch. Then, we

calculate the mean carbon value across the entire site and multiply the distribution by that number. That way we are able to obtain a constant carbon value for every tree and derive a number that approximates the volume of trees in every patch. The resulting carbon distribution is presented in Fig 7. The usage of the tree density approach produces smoother carbon distributions. The new labels more accurately correspond to the true nature of the data used for training our models.

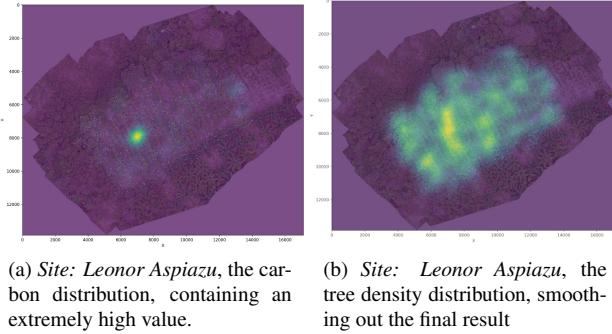


Figure 7. Comparison of gaussian and tree-density-based carbon distributions.

5. Experiments

Following the three approaches described in Sec 4 we obtain distinct datasets, each containing patches with an associated target carbon value. This represents a regression task on image data. We decide to train on 5 sites and leave one out for testing and repeat this for each site. Thus, in total, we are training 6 models for each approach. We use the ResNet18 model [5], following what the authors of [9] suggest. As a benchmark, we also recreate their pipeline but ensure that the appropriate site boundary is used. In the following, we provide more details on the general setup, present the results for the benchmark and our three approaches and lastly draw a comparison.

5.1. General Setup

We implement our code in Python, using PyTorch [7] for all neural network related tasks. We train our models on the Euler cluster of ETH [15] or locally on a MPS accelerated MacBookPro M1. Refer to A.4 for a list of neural network related hyperparameters used for all runs. Furthermore, as site boundary, we use the α -shapes as described in Sec 3.1 and shown in A.3. We settle for $\alpha = 15000$, leading to a sufficiently tight but smooth boundary. For all approaches relying on patching, we explore different combinations of patch size and rotations but do not find a significant difference in performance. We decide on a patch size of 224 and use three rotation angles for augmentation:

$0^\circ, 30^\circ, 60^\circ$. We then filter out all patches that lie more than 20 % outside of the boundary. Lastly, for the gaussian and tree density based runs, we use the gaussian prior computed in Sec 3.3 and shown in A.2. All of these hyperparameters are summarized in table A.5.

5.2. Benchmark

To compare our approaches to the pipeline presented in the ReforesTree paper, we decide to implement the latter while correcting some of their mistakes. We also replaced the AGB labels by carbon labels as they are related by a simple formula. As a starting point, we used the bounding boxes and field measurements provided in the raw dataset. For the matching algorithm, we used the classification of bounding boxes into banana and non-banana trees also contained in the data. The remaining bounding boxes were then matched to the field data points using a Sinkhorn optimal transport algorithm with greedy matching. We used the Python Optimal Transport (POT) library [4] to create a cost matrix for the transport problem in pixel coordinates. The coordinates were normalized beforehand to avoid numerical errors. The Log Sinkhorn algorithm implemented in POT was then used to compute the optimal transport plan, with regularization parameter 0.1. To achieve a one-to-one matching, the greedy matching strategy described in Sec 2.2 was used. The result of this matching is shown in Fig 8 for the site *Flora Pluas*. Many bounding boxes are matched to field measurements far away, in particular, many are matched to data points around the boundary. This is not due to too little field data in the interior. When the discarded field data is inspected, many data points which would make a better match can be found. We believe that this effect is caused by a combination of inaccurate tree crown detection and the difficulty of finding the correct boundary of the agroforestry site.

After the matching, we are left with tree crown images and corresponding carbon labels. This data was then used to train a ResNet18 model.

The results show that the model is unable to make meaningful predictions on this dataset. The combination of large outlier values with noisy matching is likely throwing off the model. As shown in Fig 9, some of the high outlier carbon values are matched to smaller bounding boxes usually corresponding to smaller trees with low carbon values. This makes it hard for the model to learn as similar trees may have very different labels. The figure also shows that the model does not predict the higher outlier values correctly resulting in under-predicted carbon across this particular site. For other sites with less outlier values, the model is over-predicting the carbon across the entire site. In summary the noise in the raw data, as well as the processing and outlier values, results in a dataset with questionable labels which prevent accurate predictions from the model.

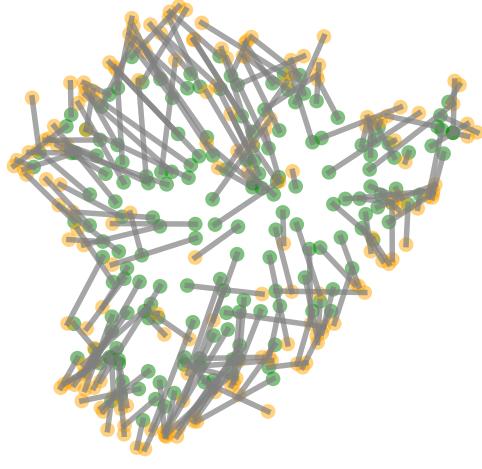


Figure 8. Matching of tree crowns (green) to field data (yellow) for *Flora Plus* site. Unmatched field data is not shown.

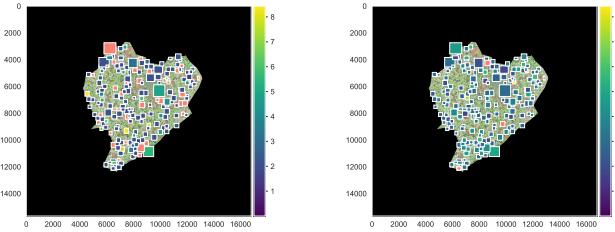


Figure 9. Labels and predictions for the *Flora Plus* site. The color grading indicates the carbon label with pink representing high outlier values.

5.3. Patching

For this approach, we used the image patches and carbon sum target obtained using the method described in Sec 4.1. The resulting target distribution, as seen in Fig 10 is inhomogeneous, with a big proportion of the patches containing zero carbon and others containing big spikes. Looking at Fig 1, it is clear that this should not be the case as the sites are covered quite evenly by trees.

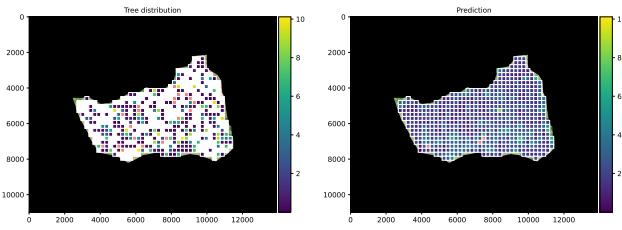


Figure 10. Labels and predictions of the patching approach for *Leonor Aspiazu* site.

5.4. Gaussian

We compute the gaussian carbon distribution as described in 4.2 using the hyperparameters described above. The training loss converges. However, the model is unable to make meaningful predictions as shown in Fig 11. It predicts noisily around the mean. Additionally, it is thrown off by the large carbon values of some trees in the training data.

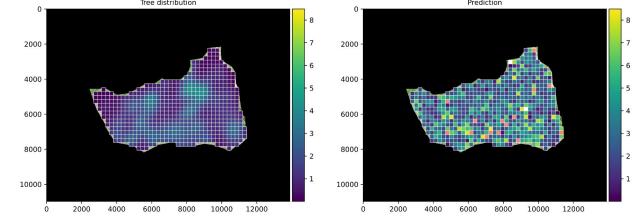


Figure 11. Labels and predictions of gaussian approach for *Manuel Macias* site

5.5. Tree Density

As presented in Sec 4.3, for this experimental setup we exploit a tree density based set of labels for training. This approach produces considerably smoother label distributions, without producing extreme dissimilarities between similar looking parts of the forestry sites. By doing so, we aim to train network architectures that are able to correctly infer the underlying distribution of our dataset, without being thrown off by outlying values that correspond to similarly looking RGB input. Unfortunately, similar to the previous pipelines proposed, including the benchmark one, the tree density approach is unable to accurately capture the ground truth distribution and produces mostly noisy results, as shown in Fig 12.

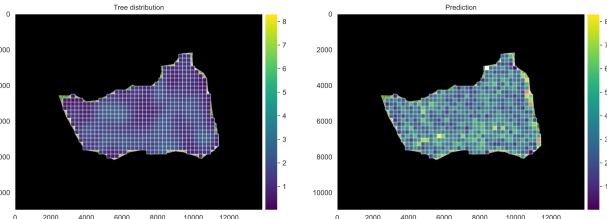


Figure 12. Labels and predictions of tree density approach for *Manuel Macias* site

5.6. Summary and Comparison

To sum up, neither the original approach of [9] nor our models are able to make meaningful predictions for the dataset. The R2 score is negative for all runs, as shown in Fig 13. This is not surprising given all the challenges and issues with the field data, outlined in Sec 3. Our attempts to

arrive at a meaningful ground truth can not undo the noise in the GPS location. However, looking at the RMSE score in Fig 13, one can see the effect of the gaussian assumption. It smoothes the distribution and leads to less variance between sites and an overall lower RMSE in the model’s predictions. In the tree density run, we assume the same carbon value for all trees. This reduces the variance and RMSE even further.

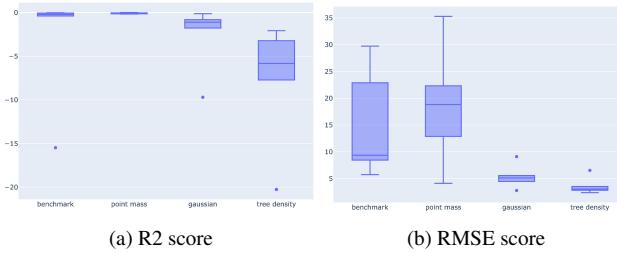


Figure 13. Comparision of all four approaches. We ran each approach 6 times: Training on 5 sites, and testing on a changing last one. Here the R2 scores and RMSE scores of these 24 runs are plotted. The R2 scores show that none of the approaches lead to meaningful predictions. The RMSE scores hint that through our gaussian and tree density approach the variance in the predictions is greatly reduced.

6. Conclusion

ReforeTree was presented as the first ever machine learning ready dataset for carbon stock estimation using drone images of tropical forests. We show that it contains several inconsistencies and issues as described in Sec 2.2 and Sec 3, such that any results obtained by training a deep learning model on this dataset are insignificant. Our approach focuses on creating a more coherent dataset rather than improving on the baseline CNN model. It simplifies the process by removing the need for an object detection model as well as a matching algorithm. Also, it quantifies and accounts for the GPS noise. Lastly, our approach makes for a smoother carbon distribution by using tree density estimates instead of carbon estimates directly.

Despite these efforts, the deep models we trained using our new datasets still fail to predict the desired carbon targets and essentially predict noise. We tried several approaches to overcome this, notably augmenting the data by using rotations, smoothing the target distribution or even using a larger ResNet50 network but to no avail. We think this phenomenon is largely due to the non-homogeneity of our data across a small number of sites. It could also be explained by our decision to split the data by site instead of randomly across sites, which was warranted by the use of rotations. In summary, we were confronted with the difficult problem of estimating forest carbon stock using drone images and very noisy and erroneous field measurements. We learned the importance of producing good and mean-

ingful data to train models on, and how big of a part this plays in any real-world application of machine learning.

Our work could be further extended by testing our proposed approaches on other existing forestry datasets. Though we already hand-labeled a few hundreds of trees, we also see the possibility to hand-label all of them and use them as the ground truth in the tree density based approach.

References

- [1] Kenza Amara, David Dao, Björn Lütjens, Dava Newman, Tom Crowther, and Ce Zhang. Towards a global species dataset by fusing remote sensing and citizen science data with graph neural networks. *Fragile Earth Workshop at KDD 2020*, 2020. 2
- [2] Gregory P Asner. Tropical forest carbon assessment: integrating satellite and airborne mapping approaches. *Environmental Research Letters*, 4(3):034009, sep 2009. 1
- [3] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29(4):551–559, 1983. 3
- [4] Rémi Flamary and Nicloas Courty. Optimal transport between 2d empirical distributions — pot python optimal transport 0.8.2 documentation, 2021. 4, 6, 10
- [5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 3, 4, 6
- [6] Alice R. Jones, Ramesh Raja Segaran, Kenneth D. Clarke, Michelle Waycott, William S. H. Goh, and Bronwyn M. Gillanders. Estimating mangrove tree biomass and carbon content: A comparison of forest inventory techniques and drone imagery. *Frontiers in Marine Science*, 6, 2020. 1
- [7] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 6
- [8] Gabriel Peyre and Marco Cuturi. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5-6):355–607, 2019. 2
- [9] Gyri Reiersen, David Dao, Björn Lütjens, Konstantin Klemmer, Kenza Amara, Attila Steinberger, Ce Zhang, and Xiaoxiang Zhu. Reforestree: A dataset for estimating tropical forest carbon stock with deep learning and aerial imagery. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(11):12119–12125, Jun. 2022. 1, 2, 4, 6, 7
- [10] Dao David Reiersen Gyri. Reforestree. <https://github.com/gyrei/ReforeSTree>, 2022. 2
- [11] Pearson Timothy; Walker Sarah; Brown Sandra. *Sourcebook for Land Use, Land-Use Change and Forestry Projects*. World Bank, Washington, DC, 2013. 1

- [12] Wanlong Sun and Xuehua Liu. Review on carbon storage estimation of forest ecosystem and applications in china. *Forest Ecosystems*, 7, 12 2020. 1
- [13] Kun Tan, Shilong Piao, Changhui Peng, and Jingyun Fang. Satellite-based estimation of biomass carbon stocks for northeast china's forests between 1982 and 1999. *Forest Ecology and Management*, 240(1):114–121, 2007. 1
- [14] Ben G. Weinstein, Sergio Marconi, Stephanie Bohlman, Alina Zare, and Ethan White. Individual tree-crown detection in rgb imagery using semi-supervised deep learning neural networks. *Remote Sensing*, 11(11), 2019. 2, 3
- [15] ETH Zuerich. Euler - scientific computing, 2022. 6

A. Supplementary Figures

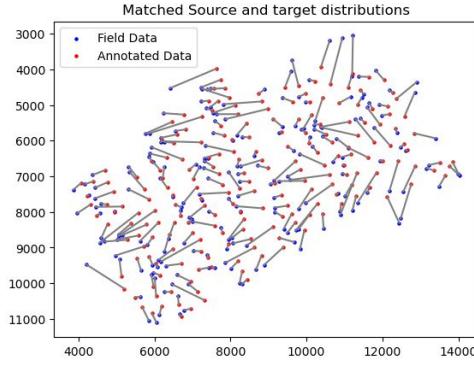


Figure A.1. Using optimal transport [4] to match field data points to target labels. Here the target labels represent hand annotated *Musacea* trees, clustered to align with the number of field data points.

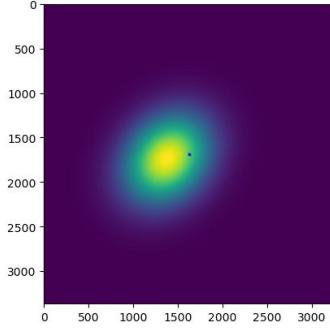


Figure A.2. Fitting a 2D Gaussian on the vectors between pairs of field data and annotated data. Assuming the matching is optimal, this represent a lower bound for the GPS error. The distribution is characterized the parameters given in A.5.

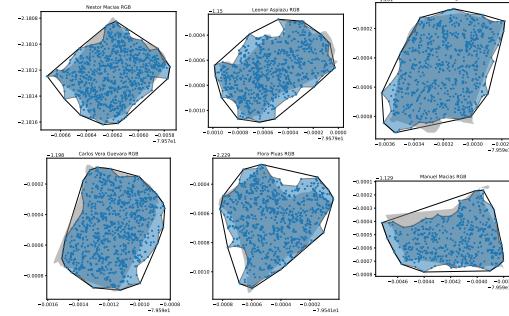


Figure A.3. Different boundaries plotted over field data points with the grey background representing the boundary provided by the dataset, the outer black polygon being the convex hull over the field data points and the blue polygon being the alpha shape over the field data points.

| Parameters | Values |
|------------------|----------|
| Learning Rate | 1e-5 |
| Number of epochs | 50 |
| Loss function | MSE Loss |
| Optimizer | amsgrad |
| Batch size | 64 |
| Normalization | Standard |

Figure A.4. Neural Network related hyperparameters used for all four models

| Parameters | Values |
|-----------------------|-----------------------------------------------------------------------------|
| α -Shape | 15000 |
| Patch Size | 224 px |
| Rotation Angles | $0^\circ, 30^\circ, 60^\circ$ |
| Filter Threshold | 20% |
| Gaussian - Mean | $[-246.4, 57.0]$ px |
| Gaussian - Covariance | $\begin{bmatrix} 106196.7, -24666.1 \\ -24666.1, 113349.2 \end{bmatrix}$ px |

Figure A.5. Data related hyperparameters used shared across some models