Forest Carbon Quantification: A Inexpensive Deep Learning Tool to Estimate Forest Canopy Height through 2D Satellite Imagery

Ian Ng
Stanford University
iyhn8192@stanford.edu

Jason Ping
Stanford University
jjping@stanford.edu

Selena Sun
Stanford University
selenas@stanford.edu

Abstract

In this project, we aim to solve the problem of estimating how much biomass is present in a region of land in one of the worlds' largest collective carbon sinks: forests. To do so, most modern techniques involve using remote sensing techniques such as LIDAR to estimate factors such as canopy height that are directly correlated to amount of biomass and therefore carbon capture. However, these methods are prohibitively expensive and take a long time. We suggest a much simpler method, which if successful can use far more available data, that of satellite images, to estimate biomass. We use two major deep learning methods – convolutional neural networks (CNNs) and generative adversarial networks (GANs) – to determine the feasibility of such an estimation. With the CNN technique, we can get as close as within 20 meters of estimation, while the GAN did not yield workable results.

1. Introduction

Consistent carbon monitoring of forests is critical in planning for climate change mitigation efforts, such as risk assessment for wildfires and conservation of forest ecosystems. Carbon stock can be estimated through above ground biomass (ABG) measurements. The most direct forms of measuring ABG include laborious and destructive methods, requiring cutting down and drying of trees to assess the dry biomass of each sample [1].

As such, current methods of quantifying carbon stock within forests rely on remote sensing techniques to capture data on forest structure to predict ABG. Of the data collected on forest structure, the forest canopy height is consistently seen to be the key to an accurate assessment of forest biomass [2, 3].

Generating a canopy height model (CHM) involves subtracting the digital terrain model (DTM) representing the elevation of the bare earth's surface from the digital surface model (DSM) representing the elevation of the earth including the environment's features, in this case the tops of trees [4]. Currently the most accurate and advanced remote sensing technique of creating the DTM and DSM elevation maps relies on Detection and Ranging (LiDAR) or airborne Interferometric Synthetic Aperture Radar (IFSAR or InSAR) technologies to acquire the necessary high-precision three-dimensional information of forests. However, both methods require the use of active sensors mounted onto aircrafts flying across survey regions, making data acquisition expensive, time-consuming [5]. Further, the generation of a DEM consists of a challenging and labor-intensive process of classifying and removing all man-made and vegetation features [6].

With the growing environmental threat caused by forest wildfires, there is an increasing need for an inexpensive yet accurate means of regularly measuring carbon stock in forests to help with forest conservation efforts and to aid governments to efficiently triage potential fire threats.

In order to address this need, we propose two deep learning tools that can create accurate canopy height models across a survey region given only satellite imagery. Specifically, the input to our model is Google Earth satellite imagery captured over forest regions in Alaska. Our model are a Convolutional Neural Network (CNN) a trained Conditional Generative Adversarial Network (cGAN) with a U-Net architecture. The model output is a canopy height map of the same region captured by the satellite images, with each pixel value representing the average canopy height across the pixel's respective geographical area.

2. Related Work

Traditionally, researchers have used LiDAR as an estimator for above-ground biomass in forests [7, 8, 9, 10]. LiDAR techniques have recently been combined with other techniques to improve accuracy, such as LiDAR collected with Unmanned Aerial Vehicles or multispectral images [11]. Despite the recent improvements, collection of LiDAR data is expensive, logistically challenging, and faces the geographic limitation of only mapping locations reach-

able by aircraft [12]. Thus, modern developments focus on using satellites to expand carbon estimation capability and resort to using LiDAR CHMs as ground truth labels. [13, 14, 15, 16, 17]

The Random Forest (RF) Algorithm is the most ubiquitous model used for generating CHMs [16, 18, 19]. They are popular for a variety of reasons: it is able to handle a wealth of input variables, and there is minimal parameter tuning needed [18]. The state of the art models are able to achieve highly accurate results (up to R^2 = 0.88) [19]. Random Forest models work best for regression tasks [18, 17, 20, 19], but dubitably well for classification tasks. Boutsoukis et al. proposed a classification task and achieved a high accuracy of $R^2 = 0.91$, but note that there are only 6 classification bins, one of which encompasses a wide range of canopy heights (5-40m) [15]. Such large bins naturally leads to high accuracies. For successful RF models, some useful practices have been acquiring LiDAR and satellite data within the same year and season [17], decomposing images into their Fourier components and using the resulting radial power spectra as features [18], and further splitting data into "young" and "mature" forest areas [19].

Another method that shows potential is a per-pixel machine-learning algorithm in the form of a regression tree [21]. This is one of the only attempts of generating a global forest canopy height map, and it has notable results ($R^2 = 0.62, RMSE = 6.6$m). Considering this a prediction over a large part of the world, the results are impressive. The authors chose to use Landsat satellite data (spatial resolution = 30m) and GEDI LiDAR data gathered on the International Space Station (spatial resolution = 25m). A potential improvement to this paper is to choose higher resolution data (e.g., Planet Dove satellite resolution is 3-5m, and the Global Airborne Observatory LiDAR data has resolution = 1.12m).

There seems to be a consensus in the carbon quantification community that most findings are region-specific. For example, Csillik et al. created a forest canopy height of all of Peru with R^2 = 0.65, capturing 128.5 million hectares of land, but hesitate to generalize to other countries [18]. Other papers contain their predictions similarly for Vancouver Island, Canada [19] and Victoria, Austrlia [20]. Garcia et al. provided a few insights: CHMs generalize best to both the location and the same forest type that the training data was gathered in [16]. Furthermore, the results counter-intuitively suggest that more coarse data could be more accurate predictors (R^2 = 0.81 at resolution = 100m, R^2 = 0.68 at resolution = 25m for the Howland location).

3. Methods

Our goal is to create a model that can successfully estimate the height of the canopy at a given point on a map, given its surroundings as viewed from a satellite image. This allows us to predict both the specific height at a particular target area, as well as create a general map out of our predictions for adjacent points. Given the success in the Random Forest tasks, we aim to take a more exploratory approach with this paper. We will be taking two different approaches to the problem. The first is the convolutional neural network approach, where we **give a convolutional neural network a 256x256 image representing a target point's surroundings, and train it to output a single number, representing the canopy height of the point at the center**. The second is the generative adversarial network. With the generative adversarial network, we will be **generating a canopy height map corresponding to the area created by our input image**.

Overall, we expect our convolutional neural network to perform better, since it's given more data per point; on the other hand, for our generative adversarial network, points near the edge will probably be less accurate, since they have less local information about their neighboring area.

Note that in all approaches, we are normalizing all images to have values that lie between 0 and 1, and scaling all scalar output values to follow a normal curve with mean 0 and standard deviation 1.

3.1. Convolutional Neural Network

As one of our main methods of approaching the problem, we use the Convolutional Neural Network, often used in image classification and regression tasks due to its versatility and ability to adapt to a wide variety of data. Our dataset is relatively large (over 20,000 256 x 256 images), and so our model has a high number of parameters and is difficult and time-expensive to train, limiting the depth and width of our model to a degree. We choose to cap the size of our model at 4 layers of depth for efficiency (3 convolutional layers and 1 fully-connected layer at the end), though with the large image size it is possible that a greater depth might be beneficial. We ablate over the depth, learning rate, and the convolutional size of each layer, allowing the convolutional network to train over a uniform of 10 epochs for over 20,000 images.

3.1.1 Loss Function

Since we are not doing multi-class classification, but regression, in which a single number is output, we are limited in our loss metrics- we can't use binary cross-entropy or cross-entropy, nor softmax. The most commonly used of the regression-suited loss functions is known as **Mean-Squared Error (MSE) Loss**. MSE Loss is the average squared difference over a batch between the model's pre-

dictions for a batch and the actual values, as follows:

$$Loss_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

where N is the size of the batch (in our case, 12), y_i is the actual value and $\hat{y}_i$ is the value predicted by our model. In addition, to prevent overfitting, we add an L_2 loss regularization term to our model:

$$Loss_{\text{reg}} = \lambda \sum_{j=1}^{k} \|w_j\|^2$$

where λ is a hyperparameter that we set at a static 10^{-3}, hence making our overarching objective function

$$Loss_{\text{tot}} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{k} \|w_j\|^2$$

3.1.2 Optimizer

For our optimizer, we used the adaptive moment-based descent algorithm Adam, which calculates the gradient updates as follows:

$$\begin{aligned} m_t &= \beta_1 \cdot m_{t-1} - (1-\beta_1)\cdot g_t \\ v_t &= \beta_2 \cdot v_{t-1} - (1-\beta_2)\cdot g_t^2 \\ \delta_t &= -\alpha \frac{v_t}{\sqrt{m_t + \epsilon}} \cdot g_t \end{aligned}$$

where α is the learning rate, g_t is the gradient, and β are tunable parameters (in practice, often 0.9 and 0.999, respectively.)

We selected Adam because it is typically generalizable to a high number of parameters and performs relatively stably in practice. Though in theory there are other optimizers (AdaGrad, RMSProp, etc.), none consistently and predictably outperform Adam in most machine learning applications.

3.2. Generative Adversarial Network

In our Generative Adversarial Network (GAN), we employ Google's Pix2Pix, a conditional generative adversarial model designed for image-to-image translation (past applications of Pix2Pix include converting outline drawings of shoes into "real" images, or layouts of storefronts into "real" stores). We are appropriating Pix2Pix and making modifications to it in order to create a conditional model that translates a satellite image to a canopy height image.

3.2.1 GAN Objective

Our GAN is optimized on an version of the cGAN, or conditional GAN, objective. In a conditional GAN, rather than generating based on purely random noise z, the generator uses random noise combined with a base image x (in this case, our satellite image), which it can be said to "condition" its generated output upon. Our objective is therefore as follows:

$$\begin{aligned} \mathcal{L}_{\text{GAN}} = \mathbb{E}_{x,y}[\log D(x,y)] + \mathbb{E}_{x,z}[\log 1 - D(x, G(x,z))] \\ + \lambda \mathcal{L}_{\text{dist}}(y, g(x,z)) \end{aligned}$$

where y is the actual output image corresponding to x. In other words, the GAN is penalized when the expected score of a discriminator is high for a real image (first term) and low for a fake image (second term). Notice the addition of a third term, $\lambda\mathcal{L}_{\text{dist}}(y, G(x,z))$. This is an extra penalization term that measures the distance between the intended output y and the generated output $G(x,z)$, which encourages the generator to converge to a more correct output.

3.2.2 GAN Architecture

Our architecture works by first downsampling an image to a deep 4x4 array, then re-up-sampling to a 256x256 array (known as a U-net architecture):

Layer	Input	Output
1 (Downsample)	256x256x3	128x128x64
2 (Downsample)	128x128x64	64x64x128
3 (Downsample)	64x64x128	32x32x256
4 (Downsample)	32x32x256	16x16x512
5 (Downsample)	16x16x512	8x8x512
6 (Downsample)	8x8x512	4x4x512
7 (Downsample)	4x4x512	2x2x512
8 (Upsample)	2x2x512	4x4x512
9 (Concatenate)	4x4x512	4x4x1024
10 (Upsample)	4x4x1024	8x8x512
11 (Concatenate)	8x8x512	8x8x1024
12 (Upsample)	8x8x1024	16x16x512
13 (Concatenate)	16x16x512	16x16x1024
14 (Upsample)	16x16x1024	32x32x512
15 (Concatenate)	32x32x512	32x32x768
16 (Upsample)	32x32x768	64x64x256
17 (Concatenate)	4x64x256	64x64x384
18 (Upsample)	64x64x384	128x128x128
19 (Concatenate)	128x128x128	128x128x192
20 (Upsample)	128x128x192	256x256x3

The U-net is designed to compress an image into a feature vector and then scale it back up into a full-size vector; we hope that this will allow the model to focus on parts of the input satellite image that are important to creating the map

(in particular, areas that seem green or have a particular texture) and scale them back up so that the textures of a canopy-height map will appear, similar to the intended output.

3.2.3 GAN Accuracy

To measure accuracy, we can simply use the average L_2 distance between the intended output and the GAN output across the image. Since each pixel of the GAN output represents the canopy height for a small area, there should be a 1:1 correspondence between the intended output and what the GAN creates as the canopy map, thereby suiting the L_2 distance. Furthermore, the L_2 is not robust to small perturbations (rotational/translational) as it penalizes offset regions the same as if they were simply incorrect; while this is not ideal for many applications of GANs that aim to be translation invariant, it is acceptable for this particular instance because our data is very location-dependent. In other words,

$$L_2 = \frac{1}{H \cdot W} \sum_{i=1}^{H} \sum_{j=1}^{W} (y_{ij} - \hat{y}_{ij})^2$$

where y_{ij} is the value at pixel ij, and $\hat{y}_{ij}$ is the predicted value at that same location.

3.2.4 Optimizer and Other Changes

As our optimizer, we again used the Adam optimizer on both the discriminator parameters and generator parameters. We ablated over the learning rate, finding our best value at $\alpha = \mathbf{2 \cdot 10^{-4}}$.

In addition, we executed a technique known as "label smoothing". Label smoothing is a means of preventing the discriminator from being overconfident; that is, preventing it from being too sure that a generated image is generated or faked, as this may lead to an imbalance between the generator and discriminator and no real learning occurring. We solve this issue by penalizing the discriminator when it makes a prediction with a confidence of over 0.9, thereby preventing it from becoming too sure of its answers and keeping both the discriminator and generator open to fluctuation. This improved our results vastly, as discussed in the "Results" section.

4. Dataset and Features

We used Google Earth data as our satellite imagery and the Interferometric Synthetic Aperture Radar (IFSAR) dataset to generate our ground truth canopy height model (CHM).

4.1. IFSAR Data

The IFSAR dataset is a collection of DTM and DSM products taken over the state of Alaska [22]. Each DTM footprint corresponds exactly to a DSM footprint, and the products have a spatial resolution of 5 meters. Airborne IFSAR data was collected from northwestern Alaska in 2012 and south-central Alaska in 2010. Since our project aims to examine forest canopy height, we selected only the DTM and DSM products from the forestry regions in south-central Alaska (covering the Tongass National Forest and Chugach National Forest) using the EarthExporer web application [23]. The region we selected covers approximately 85,000 km^2 of land.

We selected the IFSAR dataset to calculate the our ground truth CHM, which we can obtain from subtracting the DTM from the DSM [4]. The DTM and DSM data were available as TIFF files, so we converted them to numpy arrays and performed the subtraction, resulting in a CHM as a large TIFF file of approximately dimension 5800x2900.

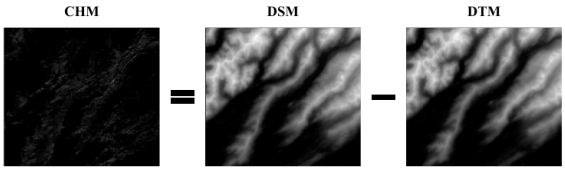


Figure 1. A sample CHM calculation (cropped to 2700x2900 pixels), where 0 - 255 meters corresponds to the black - white gradient.

To augment the data, we used a sliding window to obtain 5,300 smaller files of dimension 256x256, which we then passed into our model as labels for satellite images

.

4.2. Google Earth Satellite Imagery

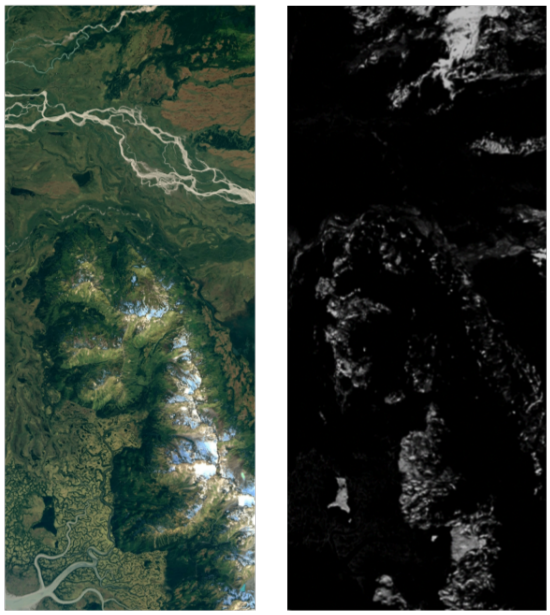
We used the Google Maps Static API to acquire satellite images of corresponding size and spatial resolution as our IFSAR data. The Maps API returns an image as a PNG file when an HTTP request via a URL is made. Matching the Google Earth satellite coordinates to the IFSAR coordinates proved to be a nontrivial challenge. To map the two geospatial datasets together, we designed our own data pipeline to capture Google Earth satellite imagery at the same spatial resolution as the IFSAR data. However, we realized the IFSAR data was captured at a slight rotation, due to the natural curvature of the Earth's surface, causing inconsistency in size and rotation between the IFSAR and Google Earth data. To correct for the rotation of the IFSAR data, we calculated the degree of rotation, counterrotated the captured Google

Earth image to match the IFSAR rotation, then cropped the Google Maps image to match the size of the IFSAR data, as described by the equations below:

$$\theta = \tan^{-1}(\frac{y_{NE} - y_{NW}}{x_{NE} - x_{NW}})$$
$$r = \frac{\|(y_{NE} - y_{NW})^2 + (x_{NE} - x_{NW})^2\|}{5 * w_{maps}}$$

where y_{NE}, y_{NW} are the latitude/longitude coordinates of the northeast and northwest corners of the original IFSAR data, and w_{maps} is the size of the originally captured Google Earth image. θ, which in practice was about 4.69^o, is the counterrotation magnitude for the Google Maps image, and r is the crop ratio (new to old).

This post-processing resulted in a PNG file of approximately size 5800x2900 pixels, which we augmented using a sliding window to obtain 5300 sub-images of size 256x256 pixels (identical to our IFSAR data post-processing).



Google Earth Satellite Image | Canopy Height Map

Figure 2. Post-processed google earth satellite image (left) and canopy height map (right).

We divided our total of 5300 labeled images into training, validation, and test sets in proportions of 80%, 15%, and 5%, respectively. The data was proportioned randomly.

5. Experiments/Results/Discussion

5.1. Baseline Model

As our baseline model, we used a net that simply took in a 256x256 image and passed it through a fully connected layer, ReLU, and MSELoss neuron.

After 50 iterations, we find that the validation MSE loss is a disappointing $\sim$ 2000, which corresponds to about a 44.7-meter mean error per image; considering that the average canopy height across all images is 16 meters high and the standard deviation of canopy height is 24 meters, this is very poor performance.

As is evident, the baseline model lacks the parameters, depth, and activations to be able to successfully model something as complex as a 256x256 geographical image to any sort of measure of success.

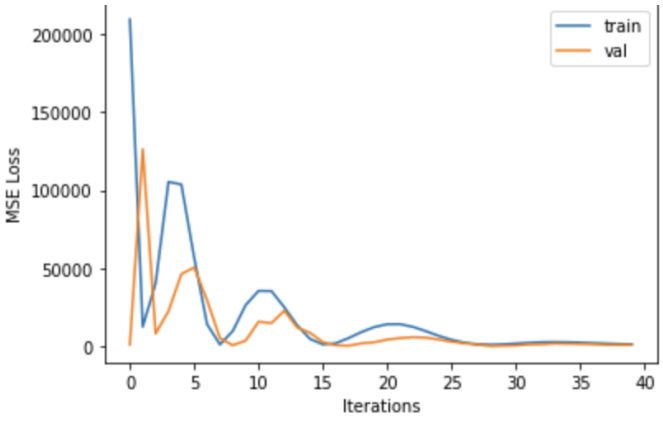


Figure 3. MSE loss graph for the FC-Net.

5.2. Fine-tuned Convolutional Neural Network

As mentioned earlier, we ablated our model over the number of layers, the size of each layer, and the learning rate of the model. We found that the best performance was found at the following learning rate and architecture:

Layer	Input
Learning rate	$1 \cdot 10^{-4}$
Number of Layers	4
Layer 1 (CONV)	256,256,3 $\rightarrow$ 128, 128, 32
Layer 1 Activation	ReLU
Layer 2 (CONV)	128,128,32 $\rightarrow$ 128, 128, 64
Layer 2 Activation	ReLU
Layer 3 (CONV)	128,128,64 $\rightarrow$ 64, 64, 32
Layer 3 Activation	ReLU
Layer 4 (FC)	64,64,32 $\rightarrow$ 1

While this leaves something to be desired in terms of size, the large dataset size makes this computationally and time expensive without the use of a pre-trained model on top of which to conduct transfer learning, and since ResNet, AlexNet, and other common pre-optimized neural networks are trained for non-adjacent classification tasks, this is unfortunately not a reasonable option.

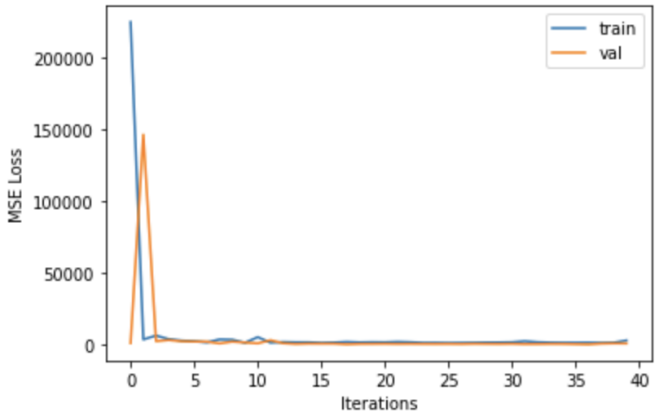


Figure 4. MSE loss graph for the fine-tuned CNN.

We can observe here that our fine-tuned convolutional neural network reaches a reasonable level of convergence much more quickly than the fully-connected net, which oscillates for some time. While this by itself isn't necessarily a good thing, it's also worth noting that after 50 iterations, the MSE loss on the validation set was around 400, corresponding to an 20-meter mean error per image. Once again, the average canopy height across all images is 16 meters, and the standard deviation is 24 meters, so this isn't necessarily a good thing. However, there are a lot of factors that may come into play here. First of all, due to the aforementioned rotation and scaling issue, not all images may line up perfectly, leading to some inconsistencies between certain data points. Secondly, there is a large amount of data provided for each target location for canopy height estimation. At a provided image size of 256x256, and pixel resolution at 5 meters per pixel, this is the equivalent of more than a square kilometer's worth of information for a single location. This abundance of information is likely to lead to overloading or confusing the model; in fact, a smaller map, with less information but information that is more localized, is likely to be beneficial for a model's overall performance on this task.

5.3. Saliency Map

We can see here that our model is in fact learning to emphasize the more important features of the image it's provided. In particular, you can see from the first saliency map that there is a focus in the gradient towards the center of the image, and that the further away a pixel is, generally the less impact it appears to have. This makes sense, considering that the target pixel we are attempting to predict the canopy height at will most likely rely more closely on the pixels nearby than the ones farther away.

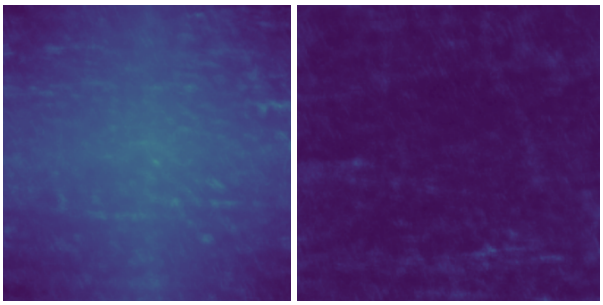


Figure 5. Left: Saliency map corresponding to a model trained to determine canopy height at the image's center. Right: Saliency map corresponding to the same model, but instead trained to determine average canopy height across the image.

To further this point, also included below is a saliency map of the same network architecture, trained to predict the average canopy height across the image instead of the canopy height at a single point; you can observe that the saliency map is much more even around the board, and merely undergoes minor fluctuations. This suggests that our model, though perhaps not numerically excellent, is at least learning to weight the correct features when determining the canopy height of the center of a given image.

5.4. Generative Adversarial Network

Unfortunately, our generative adversarial network did not yield field-viable results (loss as defined above was close to maximal value for nearly all generated images), however there was still an upward trend in terms of model performance.

In particular, we noticed that our GAN was running into a major imbalance between the discriminator loss and the generator loss; the discriminator was finding a loss of less than 1, while the generator was on the order of magnitude of 10^2. This suggests that the discriminator was too discerning with respect to the generator. To solve this issue, we implemented label smoothing, as discussed in section 3; this allowed us to fix the discrepancy between the losses, to the point where the discriminator was still significantly lower (loss of 1-2, as compared to 7-10), but at least on the same order of magnitude.

However, doing so did not solve all our problems. We ran into two distinct issues when training our GANs. The first and most important problem was that the GAN output tended to appear mostly as a gray-scale, faded version of the original satellite image. While it is true that the general shape of the canopy height map should mimic that of the original satellite image, the GAN was essentially applying light shading to the original image and producing that as its output.

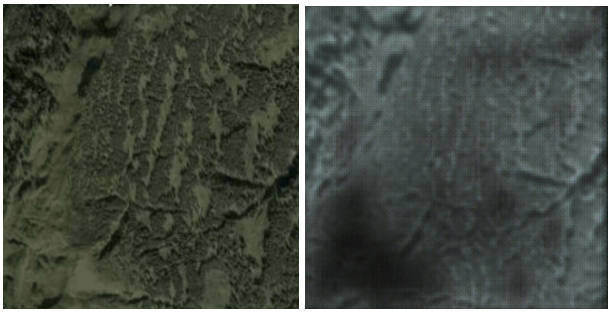


Figure 6. The prevailing issue in our GAN, where the generator simply creates a grayscale copy of the original satellite image.

The other issue was mode collapse; it would often produce the same pattern repeatedly. Both problems can possibly be solved with changes to the loss: for instance, adding a term to incorporate the Wasserstein loss, which represents how distant from the real distribution a generated image appears; this may solve the problem of mode collapse. Another possible solution is to add a loss term that penalizes too much similarity to the original image, which can encourage the generator to branch out its generated images.

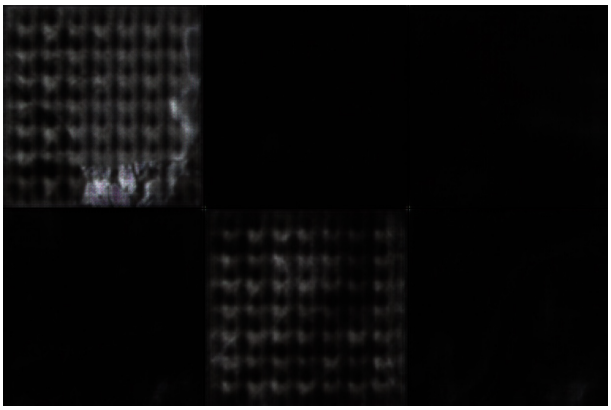


Figure 7. An example of mode collapse (top left and bottom middle figures). This pattern shows up recurringly, along with fully black images, consistently in our GAN; both are likely results of mode collapse, though the patterned image is more poignant.

Ultimately, while the GAN's results were not optimal in comparison to the non-generative models explored in this study and from others in literature, some progress was made. GAN's are notoriously difficult to train, particularly due to the delicate equilibrium necessary between the discriminator's and the generator's performance. Throughout our experience training our U-Net GAN, the same common failures occurred that often plague GANs: we experienced a discriminator that was too strong that prevented the generator from learning, the generator struggling to learn to add additional features outside of the original input image, and mode collapse due to the generator mapping the same output schemes to different input signals. However, through both careful analysis of quantitative and qualitative analysis of the respective loss scores and manual verification through visualizing output images, our team was not only able to properly diagnose these common error modes, but also successfully employed industry-standard methods in combating such issues.

6. Conclusion/Future Work

In this project, we explored two novel models to characterize carbon content in forests: a CNN and a U-Net GAN. We built our own data pipeline to collect both satellite and lidar data, match them pixel-to-pixel, and feed them into our two models. We found that our results were tenuous, but showed at least some promise, at least in the case of the CNN; the GAN requires much future work before it can avoid the problems of vanishing gradients, imbalance of power between generator and discriminator, and mode collapse. However, our results were promising in that no error mode - other than the difficulty of finding a proper equilibrium between generator and discriminator performance - was impossible to fix. This leaves room for future exploration through additional data acquisition and more computational power; our results indicate that such changes would be sure to result in a higher performing model. In future work, we'd like to explore using the ArcGIS Pro tool to build our individual CHMs into a cohesive map. We found that availability of LiDAR data (especially LiDAR point clouds that have already been built into DSMs and DTMs) are the bottleneck the locations we can run our models; given more data, we'd like to build CHMs on different parts of the world. Furthermore, we would experiment with different resolution images, since lower resolution images have been surprisingly effective in other papers. Lastly, we'd like to explore variations of the Random Forest algorithm and improve on the state-of-the-art.

7. Contributions/Acknowledgements

Ian built and tuned CNN and Jason built and tuned the U-Net cGAN. Selena built the data pipeline into the models and developed the project idea. All three members assisting in data acquisition of the satellite images and the digital elevation data. Further, preprocessing of the data - including creating canopy height models from the raw IFSAR data, registering satellite and IFSAR image samples, and generating the uniform image patches - was equally distributed. All wrote the paper and contributed equally to project progress.

Additionally, we would like to thank Dr. Derek Fong and Dr. Octave Lepinard from the Big Earth Hackathon for inspiring and guiding this project. We'd also like to thank Dr. Chris Anderson from Salo Sciences for his advice on dataset selection and model architecture.

References

[1] *Above-ground biomass estimation from LiDAR data using random forest algorithms - ScienceDirect*. URL: https://www.sciencedirect.com/science/article/pii/S1877750321001800 (visited on 06/03/2022).

[2] Cici Alexander, Amanda H. Korstjens, and Ross A. Hill. "Influence of micro-topography and crown characteristics on tree height estimations in tropical forests based on LiDAR canopy height models". en. In: *International Journal of Applied Earth Observation and Geoinformation* 65 (Mar. 2018), pp. 105–113. ISSN: 1569-8432. DOI: 10.1016/j.jag.2017.10.009. URL: https://www.sciencedirect.com/science/article/pii/S0303243417302283 (visited on 06/03/2022).

[3] Elizabeth Kearsley et al. "Conventional tree height–diameter relationships significantly overestimate aboveground carbon stocks in the Central Congo Basin". en. In: *Nature Communications* 4.1 (Aug. 2013). Number: 1 Publisher: Nature Publishing Group, p. 2269. ISSN: 2041-1723. DOI: 10.1038/ncomms3269. URL: https://www.nature.com/articles/ncomms3269 (visited on 05/08/2022).

[4] Benoît A St-Onge and Nora Achaichia. "MEASURING FOREST CANOPY HEIGHT USING A COMBINATION OF LIDAR AND AERIAL PHOTOGRAPHY DATA". en. In: (), p. 7.

[5] Marek K. Jakubowski. "Using lidar in wildfire ecology of the California Sierra-Nevada forests". en. PhD thesis. UC Berkeley, 2012. URL: https://escholarship.org/uc/item/0cj0x9hm (visited on 05/08/2022).

[6] Drew Faherty, Guy J.-P. Schumann, and Delwyn K. Moller. "Bare Earth DEM Generation for Large Floodplains Using Image Classification in High-Resolution Single-Pass InSAR". In: *Frontiers in Earth Science* 8 (2020). ISSN: 2296-6463. DOI: 10.3389/feart.2020.00027. URL: https://www.frontiersin.org/article/10.3389/feart.2020.00027.

[7] *The use of airborne laser scanning to develop a pixel-based stratification for a verified carbon offset project — Carbon Balance and Management — Full Text*. URL: https://cbmjournal.biomedcentral.com/articles/10.1186/1750-0680-6-9 (visited on 06/03/2022).

[8] Gang Shao et al. "Improving Lidar-based aboveground biomass estimation of temperate hardwood forests with varying site productivity". en. In: *Remote Sensing of Environment* 204 (Jan. 2018), pp. 872–882. ISSN: 0034-4257. DOI: 10.1016/j.rse.2017.09.011. URL: https://www.sciencedirect.com/science/article/pii/S0034425717304248 (visited on 06/03/2022).

[9] Gaia Vaglio Laurin et al. "Above ground biomass and tree species richness estimation with airborne lidar in tropical Ghana forests". en. In: *International Journal of Applied Earth Observation and Geoinformation* 52 (Oct. 2016), pp. 371–379. ISSN: 1569-8432. DOI: 10.1016/j.jag.2016.07.008. URL: https://www.sciencedirect.com/science/article/pii/S0303243416301155 (visited on 06/03/2022).

[10] Steen Magnussen, Thomas Nord-Larsen, and Torben Riis-Nielsen. "Lidar supported estimators of wood volume and aboveground biomass from the Danish national forest inventory (2012–2016)". en. In: *Remote Sensing of Environment* 211 (June 2018), pp. 146–153. ISSN: 0034-4257. DOI: 10.1016/j.rse.2018.04.015. URL: https://www.sciencedirect.com/science/article/pii/S0034425718301615 (visited on 06/03/2022).

[11] Máira Beatriz Teixeira da Costa et al. "Beyond trees: Mapping total aboveground biomass density in the Brazilian savanna using high-density UAV-lidar data". en. In: *Forest Ecology and Management* 491 (July 2021), p. 119155. ISSN: 0378-1127. DOI: 10.1016/j.foreco.2021.119155. URL: https://www.sciencedirect.com/science/article/pii/S0378112721002437 (visited on 06/03/2022).

[12] G. P. Asner et al. "High-resolution mapping of forest carbon stocks in the Colombian Amazon". English. In: *Biogeosciences* 9.7 (July 2012). Publisher: Copernicus GmbH, pp. 2683–2696. ISSN: 1726-4170. DOI: 10.5194/bg-9-2683-2012. URL: https://bg.copernicus.org/articles/9/2683/2012/ (visited on 06/03/2022).

[13] Wang Li et al. "High-resolution mapping of forest canopy height using machine learning by coupling ICESat-2 LiDAR with Sentinel-1, Sentinel-2 and Landsat-8 data". en. In: *International Journal of Applied Earth Observation and Geoinformation* 92 (Oct. 2020), p. 102163. ISSN: 1569-8432. DOI: 10.

1016/j.jag.2020.102163. URL: https://www.sciencedirect.com/science/article/pii/S030324342030026X (visited on 06/03/2022).

[14] Lauri Korhonen, Daniela Ali-Sisto, and Timo Tokola. "Tropical forest canopy cover estimation using satellite imagery and airborne lidar reference data". en. In: *Silva Fennica* 49.5 (2015). ISSN: 22424075. DOI: 10.14214/sf.1405. URL: http://www.silvafennica.fi/article/1405 (visited on 06/03/2022).

[15] Christos Boutsoukis et al. "Canopy Height Estimation from Single Multispectral 2D Airborne Imagery Using Texture Analysis and Machine Learning in Structurally Rich Temperate Forests". en. In: *Remote Sensing* 11.23 (Jan. 2019). Number: 23 Publisher: Multidisciplinary Digital Publishing Institute, p. 2853. ISSN: 2072-4292. DOI: 10.3390/rs11232853. URL: https://www.mdpi.com/2072-4292/11/23/2853 (visited on 06/03/2022).

[16] Mariano García et al. "Modelling forest canopy height by integrating airborne LiDAR samples with satellite Radar and multispectral imagery". en. In: *International Journal of Applied Earth Observation and Geoinformation* 66 (Apr. 2018), pp. 159–173. ISSN: 1569-8432. DOI: 10.1016/j.jag.2017.11.017. URL: https://www.sciencedirect.com/science/article/pii/S0303243417302763 (visited on 06/03/2022).

[17] Ovidiu Csillik et al. "Monitoring tropical forest carbon stocks and emissions using Planet satellite data". en. In: *Scientific Reports* 9.1 (Nov. 2019). Number: 1 Publisher: Nature Publishing Group, p. 17831. ISSN: 2045-2322. DOI: 10.1038/s41598-019-54386-6. URL: https://www.nature.com/articles/s41598-019-54386-6 (visited on 06/03/2022).

[18] Ovidiu Csillik, Pramukta Kumar, and Gregory P. Asner. "Challenges in Estimating Tropical Forest Canopy Height from Planet Dove Imagery". en. In: *Remote Sensing* 12.7 (Jan. 2020). Number: 7 Publisher: Multidisciplinary Digital Publishing Institute, p. 1160. ISSN: 2072-4292. DOI: 10.3390/rs12071160. URL: https://www.mdpi.com/2072-4292/12/7/1160 (visited on 06/03/2022).

[19] Oumer S. Ahmed et al. "Characterizing stand-level forest canopy cover and height using Landsat time series, samples of airborne LiDAR, and the Random Forest algorithm". en. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 101 (Mar. 2015), pp. 89–101. ISSN: 0924-2716. DOI: 10.1016/j.isprsjprs.2014.11.007. URL: https://www.sciencedirect.com/science/article/pii/S0924271614002755 (visited on 06/03/2022).

[20] Phil Wilkes et al. "Mapping Forest Canopy Height Across Large Areas by Upscaling ALS Estimates with Freely Available Satellite Data". en. In: *Remote Sensing* 7.9 (Sept. 2015). Number: 9 Publisher: Multidisciplinary Digital Publishing Institute, pp. 12563–12587. ISSN: 2072-4292. DOI: 10.3390/rs70912563. URL: https://www.mdpi.com/2072-4292/7/9/12563 (visited on 06/03/2022).

[21] Peter Potapov et al. "Mapping global forest canopy height through integration of GEDI and Landsat data". en. In: *Remote Sensing of Environment* 253 (Feb. 2021), p. 112165. ISSN: 0034-4257. DOI: 10.1016/j.rse.2020.112165. URL: https://www.sciencedirect.com/science/article/pii/S0034425720305381 (visited on 06/03/2022).

[22] Earth Resources Observation And Science (EROS) Center. *Interferometric Synthetic Aperture Radar (IFSAR) Alaska*. Type: dataset. 2018. DOI: 10.5066/P9C064CO. URL: https://www.usgs.gov/centers/eros/science/usgs-eros-archive-digital-elevation-interferometric-synthetic-aperture-radar (visited on 06/03/2022).

[23] *EarthExplorer*. URL: https://earthexplorer.usgs.gov/ (visited on 06/03/2022).