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

Ian Ng, Jason Ping, Selena Sun

Introduction

Background

Consistent carbon monitoring of forests is critical to climate change mitigation efforts and risk assessment. Carbon stock measuring of forests heavily relies on canopy height model (CHM) data.

Problem

Current means of generating CHM rely on LiDAR and IFSAR data. Both require sensors mounted on aircrafts and semi-manual data preprocessing, making regular monitoring of forests expensive and labor-intensive.

Objective

This project aimed to explore the efficacy of two different approaches towards solving this need. Specifically, we gave input data of satellite imagery to a CNN and a U-Net GAN in order to produce CHM mapped to the same geographical area. We evaluated the models using MSE and L2-distance, respectively.

Dataset

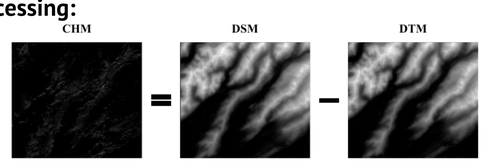
Canopy Height Model

Source: US Geological Survey's EarthExplorer

Data Description:

Digital Surface Model (DSM) and Digital Terrain Model (DTM) Interferometric Synthetic Aperture Radar (IFSAR) data spanning 85,000 km² of Alaskan forests. Resolution: 5 m² per pixel

Preprocessing:



Generating CHM from Raw IFSAR

Satellite Imagery

Source: Google Map's Static API

Data Description:

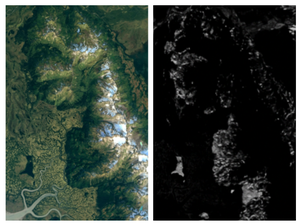
Landsat 8 satellite imagery

Preprocessing:

Correcting for IFSAR planetary rotation

$$\theta = \tan^{-1}\left(\frac{y_{NE} - y_{NW}}{x_{NE} - x_{NW}}\right)$$

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



Satellite CHM

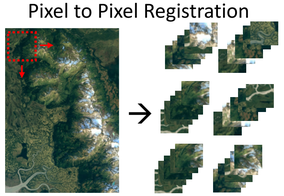
Pixel to Pixel Registration

Mini-Patches

Sliding window w/ overlap :

1 image 5800x2900 → 5300 mini-patches 256x256

Total: 23,200 patches

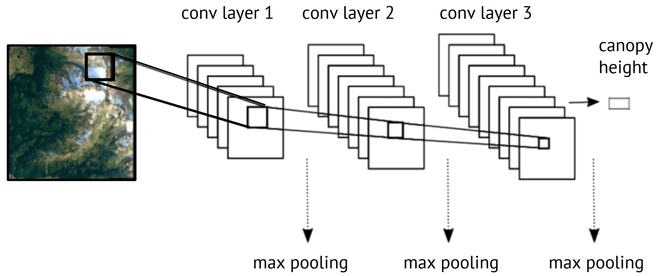


Approach 1: CNN

- We employed a 3-layer deep Convolutional Neural Network
- Input: Mini-patches of satellite imagery
- Output: Scalar value of the average canopy height of the 5 m² geographical area represented by the center pixel
- Assembling the model outputs across all pixels would allow us to generate a cohesive CHM

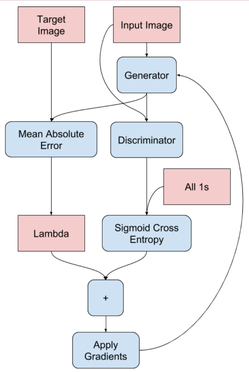
CNN Layer Dimensions

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



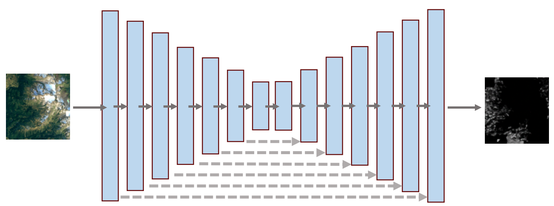
CNN Architecture

Approach 2: U-Net GAN



Training Procedure

- We employed a conditional Generative Adversarial Network that uses a U-Net as the Generator and a typical 2-layer CNN as the discriminator
- Input: 256x256 Mini-patches of satellite imagery
- Output: 256x256 Canopy Height Model
- U-Net Gan's such as Google's Pix2Pix have been seen to perform well in similar tasks that include pixel-level transformations of input images, such as style transfers

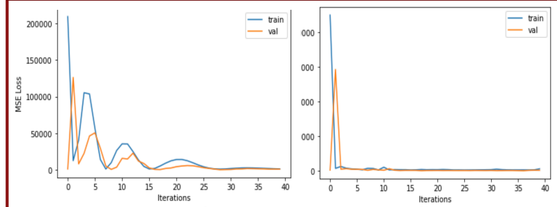


U-Net Generator Architecture

U-Net Generator Layer Dimensions

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

Results 1: CNN



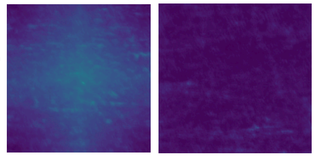
Baseline FC-net CNN

Baseline Comparison

- **Baseline (1 layer FC net):** MSE loss ~2000 → ~44.7 meter mean error per image
- **CNN:** MSE loss ~400 →~20 meter mean error per image

Saliency Maps

- We produced saliency maps for the original model (predicting height of center pixel) and one that predicted average across entire image
- Original model evidently learning to weight correct features

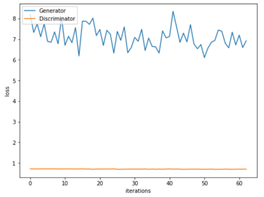


Center Average

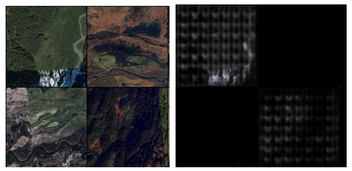
Results 2: U-Net GAN

Overpowering Discriminator

- Discriminator outperformed generator right from the beginning, not allowing the generator to learn
- Decreasing LR for Discriminator helped reduce the difference between the losses, but did not fix it



Modal Collapse

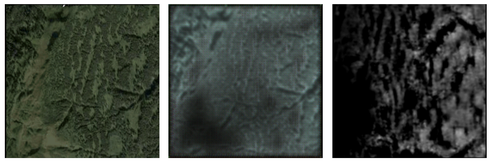


Input Output

- Produced same output but for different input images (either black or a weird grid pattern)
- This issue was fixed using label smoothing and increasing generator LR

Input Mimicry

- Our prevailing issue was that the GAN seemed to just apply a de-saturation of the input image instead of producing elevation features



Input Output Target

Conclusion & Future Steps

- Exploratory project: tested 2 novel approaches (a CNN and a GAN) at capturing 3D canopy height models through only 2D satellite imagery
- Built our own data acquisition and preprocessing pipeline, registered images pixel to pixel to match respective represented geographical areas, and improved models through diagnosis of errors and reiteration
- Results were unsatisfactory but indicated potential growth
- Next steps: Explore building CHMs into a connected, cohesive map, using LiDAR data combined or in substitute of IFSAR, and explore different resolution images