Estimating Aboveground Carbon Density in Sabah, Malaysia using Deep Learning and Sentinel-2 Satellite Imagery

Alice Zhaoyi Chen
Stanford University
alicezyc@stanford.edu

Amy Shilin Guan
Stanford University
amyguan@stanford.edu

Ryne Zen-Zhi Reger
Stanford University
rreger@stanford.edu

Abstract

Estimating carbon stock is an important step for properly understanding and preventing environmental degradation and climate change. We propose a low-cost CNN-based approach to estimating carbon stock using existing, readily available Sentinel satellite data in the state of Sabah, Malaysia. Using continuous semantic segmentation methods, we generate per pixel carbon stock predictions. We experiment with several existing regression and tree-based baselines and conduct feature importance analyses to determine that the water vapor, green, and red-edge bands from Sentinel-2 along with the Enhanced Vegetation Index (EVI) are the most important predictors. We then utilize the embeddings from the ResNet encoder as features to a shallow-CNN, and we also experiment with fine tuning pretrained segmentation model UNet. Additionally, we design a custom five convolutional layer CNN architecture with a skip connection. All layers have BatchNorm and ReLU activation, while the first three are followed by pooling for downscaling before upscaling in the last two layers. We achieve the best performance with red, green, blue, red-edge, near-infrared (NIR), and water vapor bands as input to our custom CNN, obtaining a test R^2 of 0.438, which outperforms all non-deep learning methods. Since this per pixel estimation task is essentially image generation, future work can incorporate deep learning methods made specifically for generation. Then, our model can be used to create aboveground carbon density maps for similar regions to Sabah, such as Sarawak and Kalimantan.

1. Introduction

Estimating carbon stock is important for climate change mitigation efforts, conservation and management, and for informing carbon-related and regulation policy. Quantifying carbon stock helps us understand the role of carbon sinks for climate change mitigation and better understand the impact of deforestation. It can also demonstrate the value of different forests, which leads to more conservation and biodiversity protection. With growing efforts to reduce carbon emissions, quantifying carbon stock makes potential solutions such as carbon trading and investment in carbon protection possible. These efforts are even more important in Borneo because of its rich biodiversity and the crucial role the island plays in sequestering carbon. Borneo comprises only 1% of land mass yet hosts 6% of all species according to WWF. Unfortunately, logging for economic opportunity such as oil palm has caused significant deforestation. By quantifying the carbon stock, this can lead to more sustainable economic opportunities for local communities because of the potential for investment to protect these globally-important forests.

Previous research on carbon stock estimation relies on costly and time-intensive data gathering on the ground. Researchers sample the height and width of trees and extrapolate their measurements to estimate the carbon stock of a region. Newer research reduces the time and cost by using lidar to map the height and width of trees, though this still requires tens of thousands of dollars and many labor hours. We exclusively use Sentinel-2 satellite imagery, which is freely available to all. This removes any cost and labor constraints, while also allowing for more generalizability because Sentinel-2 imagery is freely available for the entire globe. We build upon previous efforts by using the estimates from Asner et al. [2018] as our training labels.

1.1. Problem Statement

Our goal is to directly estimate carbon stock for every 30mx30m area in Sabah, Malaysia from satellite imagery. We use Sentinel-2 spectral-data and calculate additional vegetation indices from these data as our inputs. Our output is an above-ground carbon density (ACD, MGCha^{-1}) map. Specifically, our input to all following models will be a 256x256 image with channels comprised of spectral-data and vegetation indices from Sentinel-2. Our output will be a 256x256 image with a predicted value at each pixel of the ACD from the continuous range of 0-400 MGCha^{-1}.

We use linear regression, random forest, and XGBoost as

baselines. We evaluated the use of pre-trained models and finetuned a UNet model with a ResNet encoder and UNet decoder on our dataset. We also evaluated ResNet embeddings, using the embeddings of each 256x256 image as inputs for a custom shallow CNN to predict ACD. Finally, we developed a custom CNN with five convolutional layers that utilizes upsampling, downsampling, and a skip connection.

2. Related Work

There is a growing literature of efforts to use satellite imagery to estimate carbon stock. Many studies employ simple, non-deep learning models such as Random Forest and XGBoost. A smaller set use deep learning methods such as neural networks and convolutional neural networks. Furthermore, some studies combine lidar with satellite imagery, while others only use satellite imagery. There does not seem to be consensus on what is the state of the art method for carbon stock estimation because previous research all apply different methods, with none having superior results over all others in most cases. However, studies that try both non-deep learning methods with deep learning methods all find that the deep learning methods perform better.

2.1. Overview of methods

Some papers only used non-deep learning methods and were still successful in estimating ACD with satellite imagery. For example, Mngadi et al. [2021] estimated ACD in urban reforested areas with Sentinel-2 satellite imagery. With a random forest model, they achieved an RMSE of 0.378 and 0.466 and an R^2 of 79.82 and 77.96 on their calibration and validation datasets. In addition, Baloloy et al. [2018] estimated the biomass of mangrove forests in the Philippines with a combination of satellite imagery from Sentinel-2, PlanetScope, and Rapideye. With a multivariate adaptive regression spline (MARS), they achieved an R^2 of 0.89. A more comparable location to Sabah is the heavily forested Chure region of Nepal where Poudel et al. [2023] estimated aboveground biomass using only Sentinel-2 imagery. They sampled 72 plots in the region and found that their quadratic model with the normalized difference vegetation index (NDVI) performed the best with an R^2 of 0.777.

Others are starting to use deep learning approaches. To estimate forest aboveground biomass in the Huangzhou region, Tian et al. [2024] uses the Random Forest (RF), Convolutional Neural Network (CNN), and Convolutional Neural Networks Long Short-Term Memory Networks algorithms and finds that the CNN-LSTM performs the best with a R^2 of 0.74. Note that their task was to predict an estimate for each input image/plot rather than a per pixel prediction. For estimation in the Zhejiang Province with only Sentinel-2 imagery, Zhang et al. [2022] designed a custom CNN with two fully connected layers, a window size of 3x3, ReLU activation, RMSE loss, and the Adam optimizer. Their CNN achieved an R^2 of 0.7465 and 24.7745 while their linear regression model only achieved a R^2 of 0.3794. Note that they also predicted by plot/image rather than by pixel. Instead of creating their own CNN, Reiersen et al. [2022] finetuned ResNet18 to estimate carbon stock in Ecuador. Although they do not provide specific results, they note that "our baseline CNN model outperforms state-of-the-art satellite-based forest biomass and carbon stock estimates for this type of small-scale, tropical agro-forestry sites." Like the other papers, Reiersen et al. [2022] predicted by plot rather than by pixel.

Other papers used neural networks rather than CNNs. To estimate forest biomass by plot in Northeast China with lidar combined with Landsat satellite imagery, Wang et al. [2020] made a fully connected neural net with four layers, 500 neurons in each layer, ReLU activation, and dropout, which achieved an R^2 of 0.84 and an RMSE of 6.28. Csillik and Asner [2020] combined Planet Dove, Sentinel-1, topography, and lidar data to do near real-time estimation of ACD in Peru. They design a neural net with 5 layers, 3 hidden layers of 250 neurons each, ReLU activation, a linear output layer, Mean Absolute Error (MAE) loss, and the Adam optimizer, which achieves an R^2 of 0.75-0.78 and an RMSE of 20.6-22.0 for each month. Csillik and Asner [2020] is the only example of using satellite imagery to predict ACD at a per pixel level.

We hope to develop a model able to estimate carbon stock at a more precise level by predicting ACD values for each pixel.

2.2. Overview of predictors

Previous research uses a variety of inputs from lidar, field measurements, and imagery from different satellites. For the research based on satellite imagery, they all used a combination of spectral bands and vegetation indices as inputs to their model. The best predictors of ACD and biomass differed by paper. Mngadi et al. [2021] found that vegetation indices were most important. Specifically, they identified the red-edge normalized difference vegetation index (NDVIre), enhanced vegetation index (EVI), modified simple ratio index (MSR), and normalized difference vegetation index (NDVI) as the most important variables. Baloloy et al. [2018] found that the red, green, blue bands, red-edge band, NDVI, soil adjusted vegetation index (SAVI), Green NDVI (GNDVI), simple ratio (SR), and red-edge simple ratio (SRre) were most important. Tian et al. [2024], which used the CNN-LSTM found that the red-edge band and NDVI were most important, whereas Ghosh et al. [2021] determined that NDVI and GNDVI along with TNDVI were the most important.

Wang et al. [2020] found that hydro and thermo-variables were the most important. This is in line with what

Asner et al. [2018] found for what was most important for our ground truth labels—relative elevation which is "a hydrological metric related to water and nutrient availability."

3. Dataset and Features

We use the aboveground carbon density (ACD) mask for Sabah, Malaysia that was generated by Asner et al. [2018] as our "ground truth" data. This mask estimates the ACD for all of Sabah at a 30m spatial resolution and can be seen in Figure 1. One main purpose of our work is to make carbon stock estimation generalizable and accessible to all. As a result, our input data is multispectral data from Sentinel-2 accessed through Google Earth Engine and provided by Copernicus, which is accessible to all [Gorelick et al., 2017].

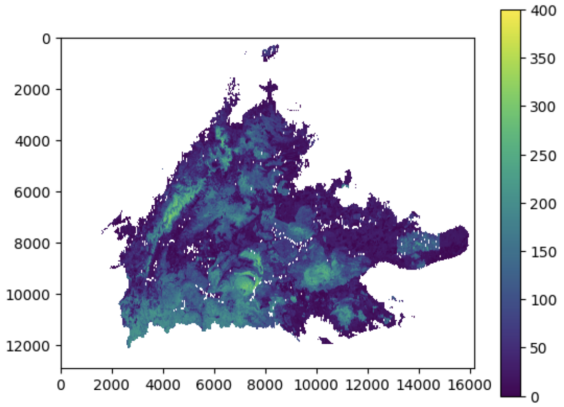

Figure 1: ACD mask generated by Asner et al. [2018], in $MGCha^{-1}$

3.1. Preprocessing

Using Google Earth Engine [Gorelick et al., 2017], we extracted images from October 15, 2015 to October 15, 2017 from Sentinel-2 to align with the date range of data gathered by Asner et al. [2018] for the mask. We filtered out all images with a cloudy percentage above 20 then took the median value of all images at each pixel for each band to form one median image. We then reprojected the median Sentinel image to align with Asner et al. [2018]'s ACD mask so each pixel of the Sentinel input aligned with each pixel of the mask where each pixel is a 30mx30m area.

We then split the image into 256x256 pixel tiles. We exclude tiles that contain over 25% NaN values, since there are several almost entirely blank regions, especially near the corners, due to the irregularity of the Sabah border (Figure 1). Out of the resulting 1106 image tiles, we then randomly sample 60% of the tiles (663 images), which we use as our primary dataset for the remainder of the project due to compute limits. We partition this into a 60%-20%-20% train-validation-test split. Since there are still a few NaN values in the labels, when we compute the loss during training, we also mask out the NaN labels so that it does not influence gradient descent, while allowing the model to retain spatial information by not simply dropping these values.

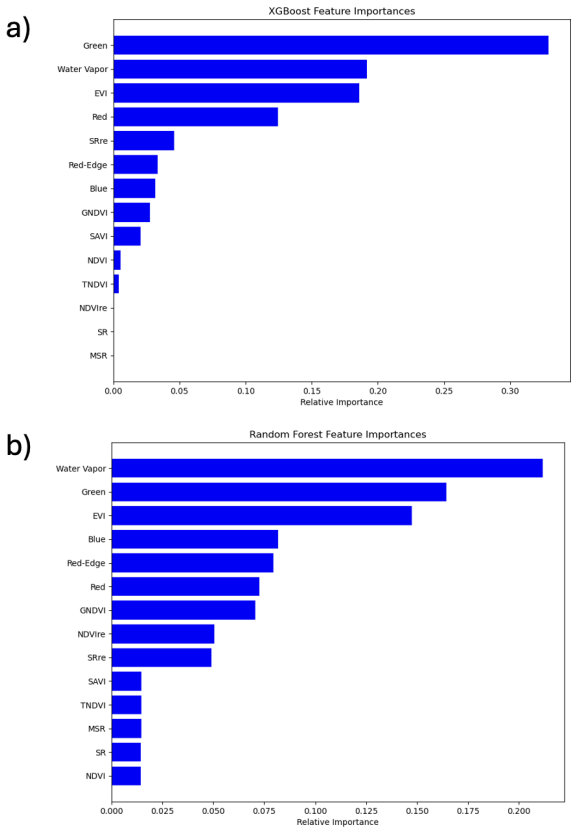

Figure 2: Feature importance tests using a) RF and b) XGBoost on all Sentinel-2 bands and vegetation indices that were deemed important by at least one previous paper

3.2. Feature Selection

To select our input channels of our models, we first turned to the literature. We run a feature importance test on every Sentinel-2 band or vegetation index that was deemed important for at least one paper's results—red, green, blue, red-edge, water vapor, NDVI, EVI, MSR, SR, SAVI, GNDVI, SRre, TNDVI, and NDVIre. We run one test using Random Forest and another with XGBoost.

As we can see in Figure 2, water vapor, EVI, and the green band are the top three most important in both the RF and XGBoost importance tests and the red-edge band is also very important in both. This aligns with the literature because Wang et al. [2020] and Asner et al. [2018] found that hydro variables were the most important. The Enhanced Vegetation Index (EVI) is made specifically to improve sensitivity in high biomass regions according to

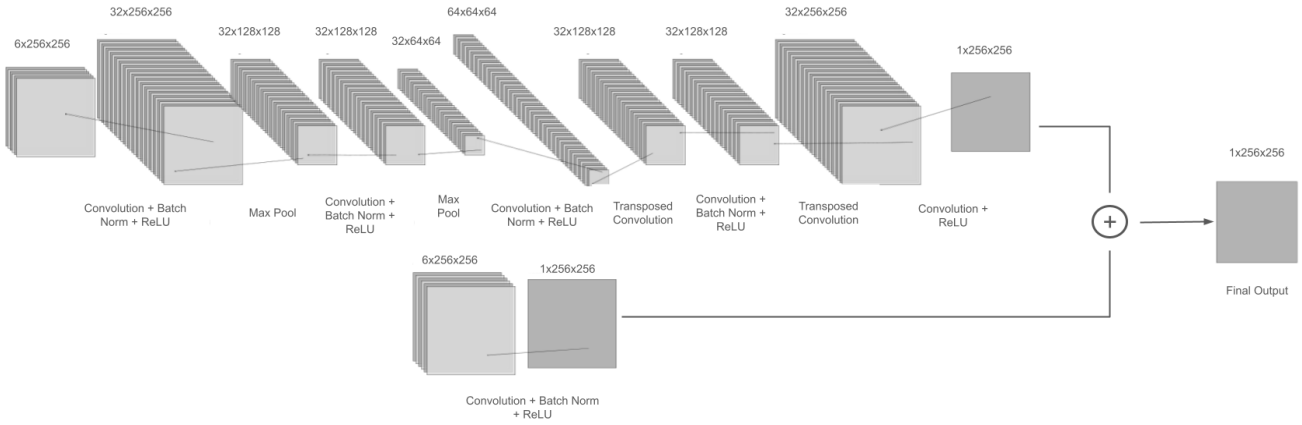

Figure 3: Custom CNN architecture with skip connection.

Jiang et al. [2008], which applies to the densely forested region of Sabah. We use the following formula for EVI with Sentinel-2 data:

$$\textbf{EVI} = 2.5 * \frac{NIR - Red}{NIR + 6 * Red - 7.5 * Blue + 1}$$

Due to the literature and our feature important results, we choose the green band (B3), red-edge band (B5), water vapor band (B9), and EVI as our four channel inputs to all of our models. We also experimented with directly using the raw inputs of EVI (blue, red, NIR) as a second set of features, including the red (B4), green (B3), blue (B2), water vapor (B9), red-edge 1 (B5), and NIR (B8) bands.

4. Methods

We use R^2 and RMSE as evaluation metrics and the mean squared error loss: $MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$ as they are used in all previous literature we have encountered. Specifically, we used the r2_score and mean_squared_error from the metrics library in scikit-learn [Pedregosa et al., 2011]. We also use accuracy as defined by Ming et al. [2021] for predicting continuous values in depth estimation, which is another regression-based per pixel computer vision task: accuracy = % of d_i s.t. $\max(\frac{d_i}{d_i^{gt}}, \frac{d_i^{gt}}{d_i}) = \delta <$ threshold , where d_i and d_i^{gt} are the predicted value at pixel i and the ground truth value at pixel i, respectively. We used a threshold of 1.25.

4.1. Baselines

We establish three baselines: a linear regression, random forest (RF), and XGBoost (XGB). We fit each baseline on two sets of features: set 1 includes the green, red edge 1, EVI, and water vapor bands, while set 2 includes the red, green, blue, red edge 1, water vapor, and NIR. For set 1, we had 3 estimators in RF and 10 estimators in XGB, and for set 2, we had 2 estimators in RF and 8 estimators in XGB. We used LinearRegression and RandomForestRegressor from scikit-learn [Pedregosa et al., 2011], and XGBoostRegressor from XGBoost [Chen and Guestrin, 2016].

4.2. Finetuning with UNet

UNet is a semantic segmentation model trained for medical computer vision tasks. UNet consists of a pre-trained encoder, such as ResNet, VGG, or others, followed by 1 x 1 bottleneck convolutional layers, and finally a custom decoder that upsamples with convolutional tranpose layers and more 3 x 3 convolutional layers, with intermittent skip connections between the encoder and decoder Huang et al. [2020]. Because this model is a widely used semantic segmentation model for satellite imagery due to its ability to handle irregular shapes [Khryashchev et al., 2018], we experiment with fine tuning UNet. After loading the pretrained model from the pytorch_segmentation_models library [Iakubovskii, 2019], we remove the multi-class classification activation in the segmentation head to apply UNet to our continuous segmentation task. After initial experimentation, and due to compute constraints, we decide to finetune the final two decoder blocks on feature set 1, the four features determined by feature importance analyses conducted during the baseline experiments. For hyperparameters, we experiment with the Adam optimizer and SGD with Nesterov momentum. Additionally, we conduct a search over learning rates of $1e-4$ to $1e-2$ and save the model with the lowest validation MSE over all pixels per sample.

4.3. Embeddings with Shallow CNN

We also experimented with using ResNet embeddings for extracting features from satellite imagery. We used

	LinReg		RF		XGB	
Bands	Train RMSE/R^2/Acc	Validation RMSE/R^2/Acc	Train RMSE/R^2/Acc	Validation RMSE/R^2/Acc	Train RMSE/R^2/Acc	Validation RMSE/R^2/Acc
Set 1	63.9/.139/.18	63.5/.132/.172	34.2/.75/.52	68.9/-.02/.174	58.6/.276/.198	59.4/.242/.186
Set 2	63.2/.158/.184	62.6/.157/.175	37.4/.706/.576	69.5/-.04/.182	58.1/.289/.199	58.8/.257/.188

Table 1: Baseline results for linear regression, random forest, and XGBoost models. Set 1 includes 4 features: green, red edge 1, and water vapor bands as well as EVI. Set 2 includes 6 features: red, green, blue, red edge 1, water vapor, and NIR. Accuracy was calculated with a threshold of 1.25.

UNet Encoder	Train Samples	Train RMSE/R^2/Acc	Validation RMSE/R^2/Acc
ResNet34	5	30.783 / 0.427 / 0.341	74.633 / -3.904 / 0.129
ResNet34	397	69.144 / -132.010 / 0.148	68.063 / -124.038 / 0.138
MobileNet_v2	397	69.039/-111142.742/0.141	68.179/-82766.547/0.131

Table 2: Results for fine tuning the last two decoder blocks of UNet on the green, EVI, water vapor, and red-edge 1 bands of our dataset. Accuracy was calculated with a threshold of 1.25. For models on 397 train samples, we use a batch size of 64. We use a learning rate of $1e-2$ for the 5 sample model, and a learning rate of $1e-3$ for the 397 sample models, with SGD with Nesterov momentum$= 0.95$.

feature set 1 (green, red edge 1, EVI, and water vapor) to generate embeddings. Specifically, we used a ResNet encoder from the segmentation_models_pytorch library [Iakubovskii, 2019], with a depth of 4 encoder blocks. This encoder generated embeddings of four sizes (CxHxW): 64x128x128, 64x64x64, 128x32x32, and 256x16x16, of which we experiment with the 64x64x64 embeddings. We then trained a shallow, custom CNN which intakes these embeddings. The shallow CNN consisted of two upsampling layers (each with a convolutional transpose 2d layer, followed by ReLU activation and BatchNorm), and a final convolutional layer to output a 256x256 ACD map. We experiment with using an Adam optimizer and SGD with Nesterov Momentum, and also conduct a search over learning rates of $1e-4$ to $1e-2$. We save the model with the lowest validation MSE over all pixels per sample.

4.4. Custom CNN

We also design our own custom CNN architecture, as shown in Figure 3. With the idea that we first want to identify basic features of the image, we downsample using convolutional layers with BatchNorm and ReLU activation. We also reduce the dimensions in each layer using pooling to combat overfitting. We then upsample with transposed convolutional layers with BatchNorm and ReLU activation to return the image to the original resolution so that we can predict a value for each pixel. Essentially, we are generating a predicted image. We add a skip connection to allow the model to reuse basic features from the start and combine them with the more complex features developed later in the net.

5. Results

5.1. Baseline

As seen in Table 1, all baseline models achieved comparable performances when fitted on the two different sets of features. While the LinReg and XGB models performed similarly on the train and validation sets, RF had severe overfitting likely because of our limit of 2-3 estimators due to compute constraints. The best baseline model was XGBoost with 10 estimators using the first set of features (green, red edge 1, water vapor, EVI), and it achieved a validation RMSE and R^2 of 59.4 and .242, respectively.

5.2. UNet Finetuning

To verify model applicability, we initially attempted to overfit the finetuned UNet model with the ResNet34 encoder on 5 training samples with the green, water vapor, and red-edge 1 bands and EVI as features. While the training loss (MSE) did not decay to 0, we did notice the train loss rapidly decrease, then plateau to an RMSE of 30.783, an R^2 of 0.427, and an accuracy (threshold 1.25) of 0.341 (Table 2). As expected, this overfit model had poor performance on the validation set, with an RMSE of 74.633 and a negative R^2 that indicated no correlation.

We then moved onto finetuning UNet with the ResNet34 encoder on a larger training dataset of 397 image tiles. After evaluation, we see that there is an overall poor fit on both the training and validation sets, with a train and validation RMSE of 69.144 and 68.063, respectively (Table 2). Especially given the higher training loss than validation loss, these metrics seemed to indicate little learning occurring, so we instead test out a separate, more lightweight

Train Samples	Train RMSE/R^2/Acc	Validation RMSE/R^2/Acc
5	50.0 / -12.361 / .184	41.8 / -4.249/ .169
20	59.6 / -5.356 / .230	63.2 / -4.884 / 0.281
397	67.8 / -44.6334 / .148	67.4 / -45.174 / .275

Table 3: Results for custom shallow CNN with ResNet embeddings inputs: overfitting on 5 and 20 samples, and training on the entire dataset. Accuracy was calculated with a threshold of 1.25. For the model on 397 train samples, we use a batch size of 20. We use a learning rate of $1e-3$ for all models, with the Adam optimizer.

encoder: MobileNet. Again, we achieve similar results, with a train and validation RMSE of 69.039 and 68.179, respectively (Table 2). These results closely mirrored those achieved with ResNet34, indicating that perhaps this specific encoder-decoder architecture may not be well-suited to our task. We hypothesize that this may be due to the fact that both ResNet and MobileNet are trained on ImageNet, which contains significantly different images content-wise than our geospatial images.

5.3. ResNet Embeddings with Shallow CNN

For our shallow CNN model with embeddings, we also first attempted to overfit our model to 5 training examples. While we observed the MSE loss decrease substantially and plateau during training, it never decayed to 0, and after 500 epochs and conducting our search over learning rates, we were only able to achieve a train RMSE of 50.0 and an R^2 of -12.3: another negative R^2 which indicates no correlation, even on the training set. We repeated this experiment on 20 samples, (which yielded train and validation RMSE of 59.6 and 63.2 respectively), and the entire dataset of 397, (which yielded train and validation RMSE of 67.8 and 67.4). All experiments yielded negative R^2 values, indicating that our model had a worse fit than a constant function set at the mean value of the dataset, and that minimal learning occurred (Table 3).

Given the consistently poor results from using ResNet as an encoder in this shallow CNN and in the fine-tuned UNet, we decided to investigate our hypothesis that ResNet is unable to generalize from standard images to satellite data. To do so, we manually inspected the ResNet embeddings of the satellite imagery using a standard PNG image as a control (Figure 4). In Figure 4a, each channel of the control photo's embedding detected a unique texture or edge feature from the image. On the other hand, Figures 4b and 4c display two tiles from the train dataset (four channels: green, red edge 1, water vapor, and EVI) and their embeddings. The ResNet embeddings fail to detect any relevant features in the original satellite imagery (such as the river present in subfigure b). Also, the different channels of the embedding appear to copy each other, generating only a few unique 'channels' per embedding. Thus, it is clear that ResNet is insufficient to extract rich features from our data, confirming our hypothesis that ResNet does not generalize well to satellite imagery. As a result, we did not proceed with trying to use other embeddings of different sizes, finetuning UNet on the other set of features, or using embeddings calculated with the second set of features to re-train our shallow CNN.

5.4. Custom CNN

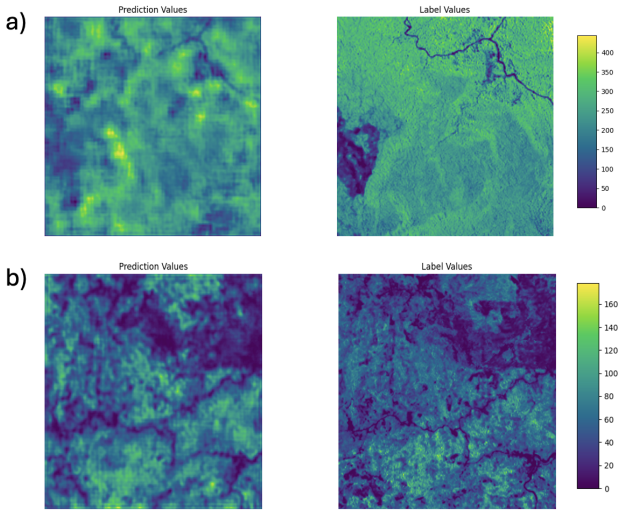

Figure 5: Predictions of our best custom CNN model (left) compared to the true ADC values (right) on two tiles in the test set in subfigures a) and b).

We trained our five convolutional layer CNN with a skip connection on both sets of features, with results shown in Table 4. With our original selection of four features (green, red edge 1, water vapor, and EVI), our model did not perform well, and obtained more negative train and validation R^2 values of -53.719, -75.817. We hypothesized that the EVI was not retaining enough information from the bands, so we separated the bands used to calculate EVI—red, blue, NIR—and ran our model with our other set of six features (red, blue, green, red edge 1, water vapor, and NIR). Our custom CNN outperformed all previous experiments with these input channels, ultimately obtaining a R^2 of 0.459, 0.435, and 0.438 on our train, validation, and test sets, respectively. Our model generalizes well as proven by its high R^2 on the test set of 0.438, which is slightly higher than the validation R^2 of 0.435.

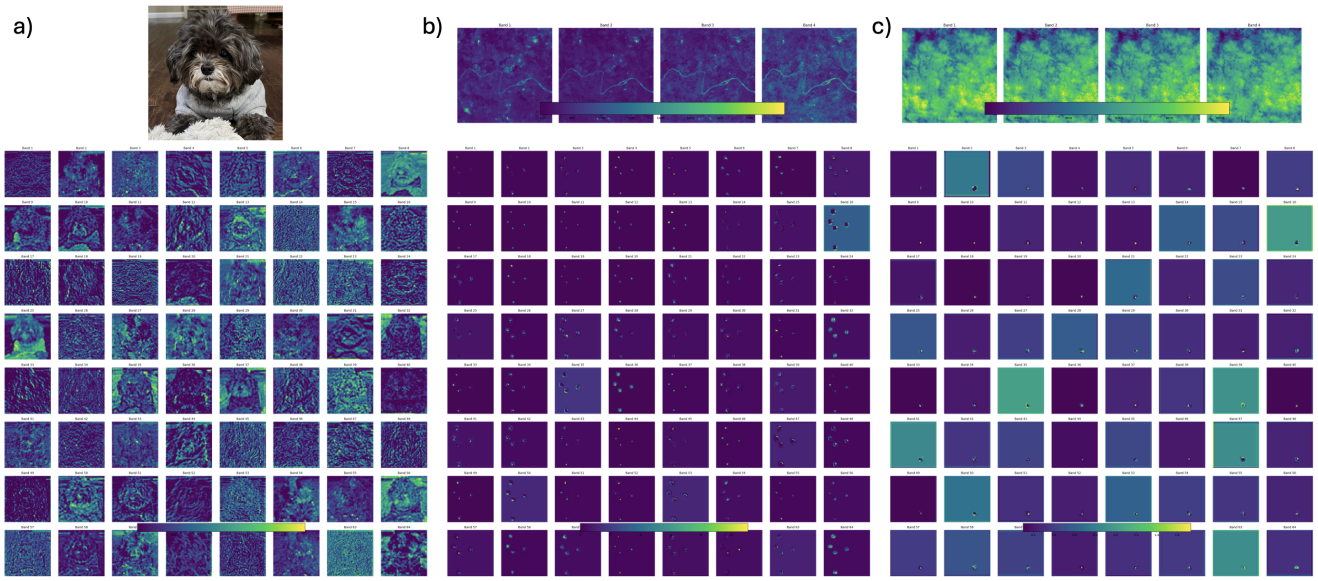

Figure 4: Manual inspection of 64x64x64 embeddings. The top images are the original inputs, and the bottom images correspond to each channel of the embeddings. a) Control photo of dog (Coal), where different channels detected different texture and edge features. b, c) 4 channels of tiles from train dataset, from which embeddings do not extract notable features.

We examine the predicted ACD masks for a few test tiles in Figure 5. The model is able to detect features that are clearly distinct from one another. For example, in tile B, the maneuvering river is appropriately detected and the model correctly predicts the denser forest in the bottom left region. However, the model struggles more when the features are less differentiable. Notice in labels for tile A how most of the image is about the same value of around 350. Our model still correctly identifies some main features such as the river, more barren area in the left, and the high carbon area near the river, but it does so with less precision than in tile B. We also see that it incorrectly predicts hot spots such as the bright yellow, curved area in the middle left. In all, our custom CNN can accurately detect and predict key features such as rivers and dense forest but struggles with precision when there is less obvious differentiation.

6. Conclusion/Future Work

We used Sentinel-2 imagery to estimate the aboveground carbon density (ACD) for every 30mx30m area in Sabah. We first tried finetuning a pretrained UNet and using embeddings trained on ResNet in a shallow CNN. These efforts were unable to produce useful results, which we determined was due to how the embeddings did not capture key features of our data. We demonstrated how the embeddings capture key features of an image of a dog, yet were unable to identify any features from the images in our dataset. We believe this is because the embeddings for both the UNet (which used a ResNet encoder) and ResNet were trained on the ImageNet dataset, which consists of images unrelated to a satellite image of a dense, rural forest in Sabah. As a result, we built a custom CNN with five convolutional layers with batchnorm and ReLU. We downsample with pooling in the first three layers then upsample and add a skip connection. Our best model was our custom CNN, which used the red, green, blue, red-edge, NIR, and water vapor bands from Sentinel-2 and achieved a R^2 of 0.459, 0.435, and 0.438 on our train, validation, and test sets, respectively. This performed better than our best non-deep learning model, which was XGBoost with a R^2 of 0.289, 0.257, and 0.249. We see that our predicted images correctly detect key features such as rivers and denser forest, demonstrating learning and ability of prediction.

6.1. Future Work

Our goal was to estimate the ACD for each 30mx30m area in Sabah. This is a significantly more complex task than estimating ACD for an entire region. This essentially makes our task a generation task, where we create a new image showing the mapping of ACD. In the future, we may want to simplify the task into one of estimation. For example, instead of 256x256 = 65536 predictions for each input image, we make one prediction for each image that represents the estimated ACD for that entire area. To improve upon our current objective, we can apply models specifically made for generation of satellite imagery such as Dif-

Bands	Train RMSE/R^2/Acc	Validation RMSE/R^2/Acc
Set 1	62.9 / -53.719 / .123	64.8 / -75.817 / .165
Set 2	40.9 / .459 / .309	41.4 / .435 / .295

Table 4: Results for custom five-layer CNN. Set 1 includes: green, red edge 1, and water vapor bands as well as EVI. Set 2 includes: red, green, blue, red edge 1, water vapor, and NIR. Accuracy was calculated with a threshold of 1.25. Both networks were trained with batch sizes of 32, with a learning rate of $1e-3$ using the Adam Optimizer. The model trained on set 1 trained for 750 epochs, and the model on set 2 trained for 1000 epochs.

fusionSat by Khanna et al. [2024]. We can also continue to finetune our existing custom CNN architecture and try different input band and vegetation index combinations. Then, we want to apply our model to other comparable regions, such as generating per pixel ACD predictions for the entire island of Borneo.

7. Contributions and Acknowledgements

Link to segmentation-pytorch-models library.

- Amy: Data preprocessing. UNet finetuning. Setting up CNN architecture and training loop. VM setup.
- Alice: Baselines. UNet setup. Embeddings + shallow CNN model. Custom CNN finetuning. VM setup.
- Ryne: Project scope, data preprocessing, baselines + feature importance selection, custom CNN design.

References

Gregory P Asner, Philip G Brodrick, Christopher Philipson, Nicolas R Vaughn, Roberta E Martin, David E Knapp, Joseph Heckler, Luke J Evans, Tommaso Jucker, Benoit Goossens, et al. Mapped aboveground carbon stocks to advance forest conservation and recovery in malaysian borneo. *Biological Conservation*, 217:289–310, 2018. 1, 3

A. B. Baloloy, A. C. Blanco, C. G. Candido, R. J. L. Argamosa, J. B. L. C. Dumalag, L. L. C. Dimapilis, and E. C. Paringit. Estimation of mangrove forest aboveground biomass using multispectral bands, vegetation indices and biophysical variables derived from optical satellite imageries: Rapideye, planetscope and sentinel-2. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-3:29–36, 2018. doi: 10.5194/isprs-annals-IV-3-29-2018. URL https://isprs-annals.copernicus.org/articles/IV-3/29/2018/. 2

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939785. URL http://doi.acm.org/10.1145/2939672.2939785. 4

Ovidiu Csillik and Gregory P. Asner. Near-real time aboveground carbon emissions in peru. *PLOS ONE*, 15(11), Nov 2020. doi: 10.1371/journal.pone.0241418. 2

S.M. Ghosh, M.D. Behera, B. Jagadish, A.K. Das, and D.R. Mishra. A novel approach for estimation of aboveground biomass of a carbon-rich mangrove site in india. *Journal of Environmental Management*, 292:112816, 2021. ISSN 0301-4797. doi: https://doi.org/10.1016/j.jenvman.2021.112816. URL https://www.sciencedirect.com/science/article/pii/S0301479721008781. 2

Noel Gorelick, Matt Hancher, Mike Dixon, Simon Ilyushchenko, David Thau, and Rebecca Moore. Google earth engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment*, 2017. doi: 10.1016/j.rse.2017.06.031. URL https://doi.org/10.1016/j.rse.2017.06.031. 3

Huimin Huang, Lanfen Lin, Ruofeng Tong, Hongjie Hu, Qiaowei Zhang, Yutaro Iwamoto, Xianhua Han, Yen-Wei Chen, and Jian Wu. Unet 3+: A full-scale connected unet for medical image segmentation. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1055–1059, 2020. doi: 10.1109/ICASSP40776.2020.9053405. 4

Pavel Iakubovskii. Segmentation models pytorch. https://github.com/qubvel/segmentation_models.pytorch, 2019. 4, 5

Zhangyan Jiang, Alfredo Huete, K. Didan, and Tomoaki Miura. Development of a two-band enhanced vegetation index without a blue band. *Remote Sensing of Environment*, 112:3833–3845, 10 2008. doi: 10.1016/j.rse.2008.06.006. 4

Samar Khanna, Patrick Liu, Linqi Zhou, Chenlin Meng, Robin Rombach, Marshall Burke, David Lobell, and Stefano Ermon. Diffusionsat: A generative foundation model for satellite imagery, 2024. 8

Vladimir Khryashchev, Leonid Ivanovsky, Vladimir Pavlov, Anna Ostrovskaya, and Anton Rubtsov. Comparison of different convolutional neural network architectures for satellite image segmentation. In *2018 23rd Conference of Open Innovations Association (FRUCT)*, pages 172–179, 2018. doi: 10.23919/FRUCT.2018.8588071. 4

Yue Ming, Xuyang Meng, Chunxiao Fan, and Hui Yu. Deep learning for monocular depth estimation: A review. *Neurocomputing*, 438:14–33, 2021. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2020.12.089. URL https://www.sciencedirect.com/science/article/pii/S0925231220320014. 4

Mthembeni Mngadi, John Odindi, and Onisimo Mutanga. The utility of sentinel-2 spectral data in quantifying aboveground carbon stock in an urban reforested landscape. *Remote Sensing*, 13(21), 2021. ISSN 2072-4292. doi: 10.3390/rs13214281. URL https://www.mdpi.com/2072-4292/13/21/4281. 2

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 4

Ananta Poudel, Him Lal Shrestha, Niraj Mahat, Garima Sharma, Sahara Aryal, Rupesh Kalakheti, and Basanta Lamsal. Modeling and mapping of aboveground biomass and carbon stock using sentinel-2 imagery in chure region, nepal. *International Journal of Forestry Research*, 2023:1–12, May 2023. doi: 10.1155/2023/5553957. 2

Gyri Reiersen, David Dao, Björn Lütjens, Konstantin Klemmer, Kenza Amara, Attila Steinegger, 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. doi: 10.1609/aaai.v36i11.21471. URL https://ojs.aaai.org/index.php/AAAI/article/view/21471. 2

Xin Tian, Jiejie Li, Fanyi Zhang, Haibo Zhang, and Mi Jiang. Forest aboveground biomass estimation using multisource remote sensing data and deep learning algorithms: A case study over hangzhou area in china. *Remote Sensing*, 16(6), 2024. ISSN 2072-4292. doi: 10.3390/rs16061074. URL https://www.mdpi.com/2072-4292/16/6/1074. 2

Xiaoyi Wang, Guanting Lv, Guishan Cui, and Jinfeng Xu. A deep learning based estimate of aboveground forest carbon density in northeast china. *ESS Open Archive*, Sep 2020. doi: 10.1002/essoar.10503989.1. 2, 3

WWF. URL https://wwf.panda.org/discover/knowledge_hub/where_we_work/borneo_forests/. 1

Fanyi Zhang, Xin Tian, Haibo Zhang, and Mi Jiang. Estimation of aboveground carbon density of forests using deep learning and multisource remote sensing. *Remote Sensing*, 14(13), 2022. ISSN 2072-4292. doi: 10.3390/rs14133022. URL https://www.mdpi.com/2072-4292/14/13/3022. 2