Evaluating Forest Density Prediction Models

Walden Marshall

May 2022

1 Introduction

It is often desired in the environmental sciences to be able to identify the amount of vegetation in a geographical region. In-situ measurement techniques have the potential to provide more accurate results than remote sensing estimates. However, in-situ measurements are inefficient in that it may take a long time to cover only a small area. For example, in-situ forest density estimates involve measuring many distances between trees to find the average distance[1]. This is why it is preferable to estimate forest density using drone or satellite imagery.

Many models for estimating forest density from imagery involve primitive hand-engineered features often referred to as indices. These are calculations performed on a per-pixel basis using the data from multiple spectral bands.

In this work, we evaluate the ability of these indices to accurately predict forest density from satellite imagery. We also generate two neural network models to predict forest density and compare performance of these various models.

2 Methods

2.1 Data Selection

In order to train a neural network, it is necessary to have feature data (in this case simply image data) and corresponding ground truth data (in this case forest density measurements). As mentioned earlier, in situ measurement of forestry is highly tedious, so we were unable to find in-situ groundtruth data. Instead we treat Copernicus'[2] 2015 estimated tree density as a groundtruth despite the fact that it is actually derived from remotely sensed data itself. So while there is potentially deviation in the Copernicus data set from the actual physical quantity being estimated, our process for developing predictive models would be the same regardless.

The Copernicus data used was derived from 2015 Sentinel-2 imagery. One tile (32TNS) located in central Europe was selected and the data was retrieved from the Sentinel Website[3]. An example of one band of the raw Sentinel-2 imagery and the groundtruth data is depicted below.

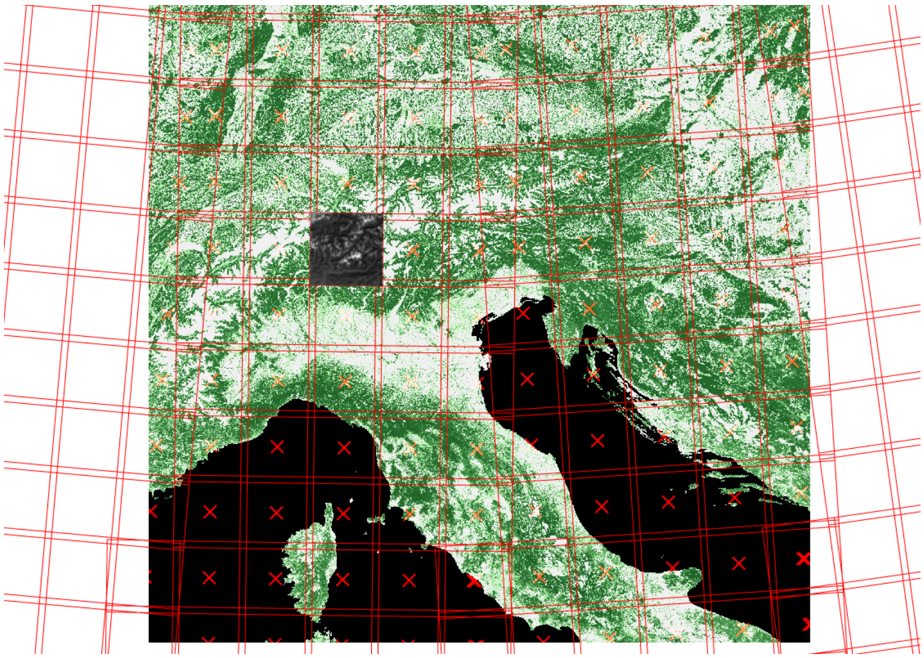

Figure 1: Sentinel-2 ultra blue band (greyscale square), and groundtruth forest density (green)

2.2 Data Preprocessing

For the neural networks, we needed a set of pixels with band data and a set of ground truth pixels corresponding to the same geographic of locations as the feature data. While the ground truth image files were not aligned with the raw satellite images, both sets of data included spatial orientation metadata. This allowed us to use ArcGIS to find the overlapping area between images and create satellite imagery and groundtruth imagery with the same geographic boundaries. Shown below is the final extracted image to be used as groundtruth.

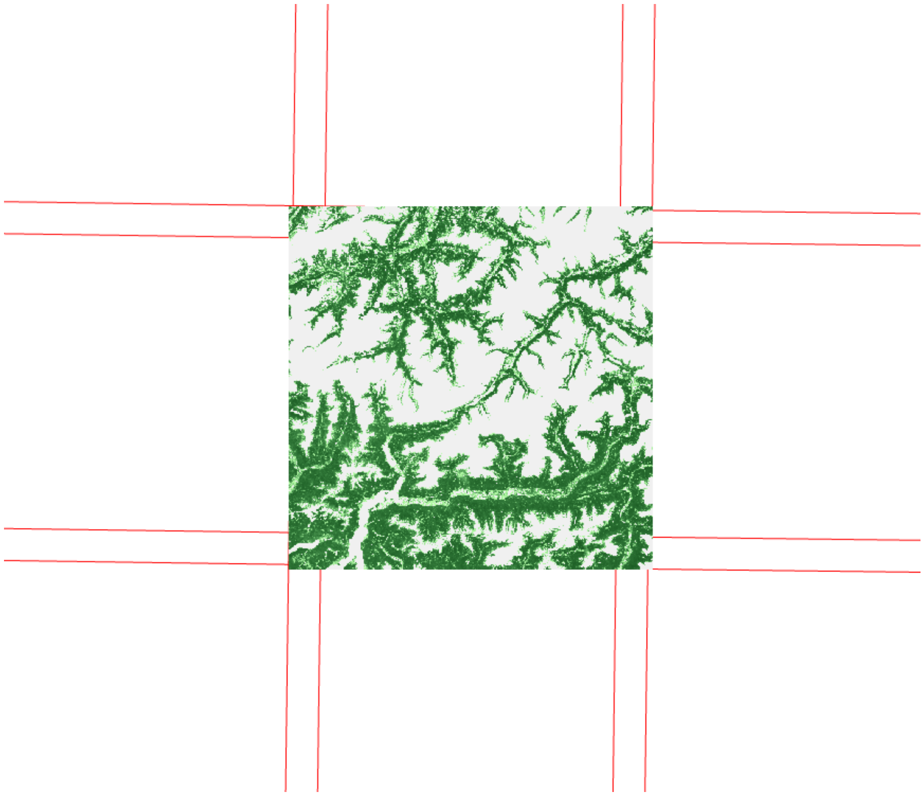

Figure 2: Groundtruth overlap with 32TNS Sentinel-2 band data

Once the images had the same spatial size, we had to create a one-to-one pixel mapping between them. This meant using Matlab to downsample the satellite imagery to have the same pixel resolution as the groundtruth data.

2.3 Machine Learning Models

We develop two neural network models to predict forest density: *BandMdl*, which takes as its input the raw satellite data, and *MetrMdl*, which takes as its inputs a number of commonly used vegetation indices. The indices chosen were some used in previous classwork: NDVI, SR, MSR, MNDVI and TSAVI (with $s = 0.33$, $a = 0.5$, $X = 1.5$). The indices are defined as follows:

$$\mathrm{NDVI} = \frac{\mathrm{NIR} - \mathrm{R}}{\mathrm{NIR} + \mathrm{R}}$$

$$\mathrm{SR} = \frac{\mathrm{R}}{\mathrm{NIR}}$$

$$\mathrm{MSR} = \frac{\mathrm{R}}{\mathrm{G}}$$

$$\text{MNDVI} = \frac{\text{G} - \text{R}}{\text{G} + \text{R}}$$

$$\text{TSAVI} = s \cdot \frac{\text{NIR} - s\text{R} - a}{a\text{NIR} + \text{R} - as + X(1 + s^2)}$$

Sentinel-2 collects data in 13 bands. However, due to limitations in computational power, some bands had to be removed. We chose to remove NIR1, NIR2, NIR3, NIR5, SWIR1, and SWIR2 because these are not commonly used in vegetation indices[4] which means they are likely not useful information for forest density prediction.

While the two networks, *BandMdl* and *MetrMdl*, take different input data sets, their architectures are the same. Both use three hidden layers of size 28, 14, and 7, "ReLU" activation functions, and all Matlab defaults for training.

The final images were roughly 1 megapixel, but due to computational power constraints, we only trained on a subset of 65000 randomly selected pixels. Due to potential divide-by-zero errors in the the indices, we ensured that the selected data subset included no values of NaN or Inf.

In order to test the performance of each index and the two neural networks, we chose another subset of 65000 pixels. This set was chosen with the stipulation that there could be no overlap between the test and training data sets.

3 Results

For the test dataset, we evaluate the mean-square-error between each index and the groundtruth forest density. Since the groundtruth data has range $[0, 255]$ and the indices have various ranges, the groundtruth data and indices' predictions were all normalized to the range $[0, 1]$.

Using the neural networks, we form forest density predictions for the test dataset. Since the neural networks were trained on input data in the range $[0, 255]$, their predictions will also be approximately in the range $[0, 255]$. Since they are nonlinear functions, we normalized the neural networks' predictions rather than the inputs to the neural network. This allowed us to directly compare the neural networks' performance to the indices. All models' predicition performances are shown below:

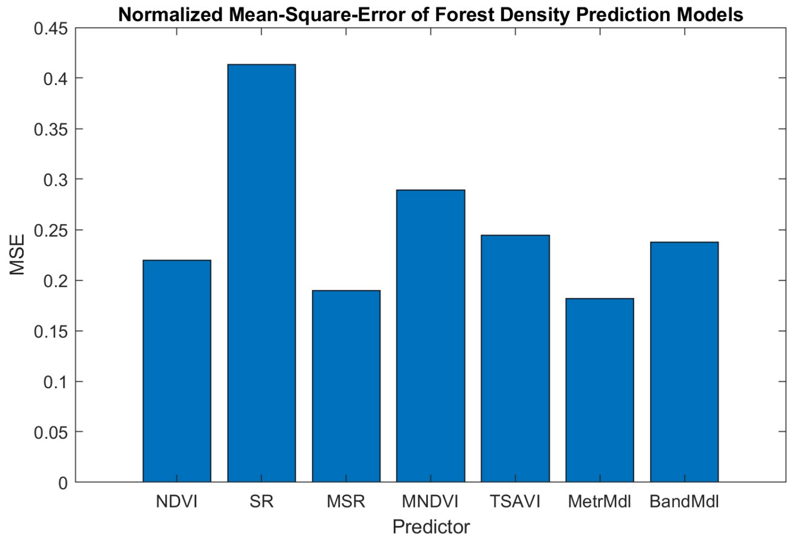

Figure 3: Performance of indices and neural networks

.

MetrModel is the most effective forest density predictor with the lowest mean-square-error of 0.1819, just outperforming MSR with a mean-square-error of 0.1899.

We certainly expect *MetrModel* to outperform all the indices since it is an optimized function of the indices. It is surprising that its margin of outperformance is so slim though.

There are several reasons why it may not have performed as well as would be expected. First and foremost, could be an error in preprocessing. While the groundtruth data from Copernicus was derived from the same Sentinel-2 data we use, Copernicus must have post-processed their data in a way which removed some of the original formatting. This resulted in an angular deviation between the geographic orientations of the groundtruth and the Sentinel-2 gridding system (see below).

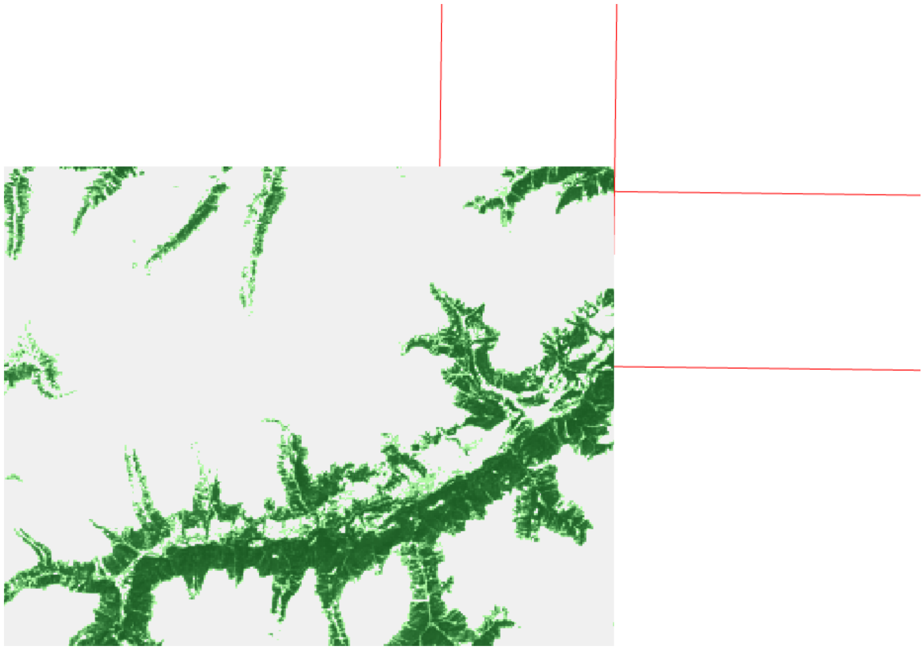

Figure 4: Imperfect alignment between Sentinel-2 grid and groundtruth imagery
.

This displacement is problematic since it means pixels in the groundtruth data won't be located at the same exact geographic location as their corresponding groundtruth pixels. While this would be very problematic for some data sets, forest density is not likely to vary by extreme amounts from pixel to pixel. This means even if some groundtruth pixels actually correspond to a neighbor's spatial location, their forest density is likely to be quite similar to their neighbors, so we are not training the neural networks on significantly distorted data.

The most likely reason for the *MetrMdl*'s underperformance is that the network and training dataset being used are both quite small due to time and computational restraints. Since the focus of this project is more on the environmental science than the machine learning, no time was devoted to hyperparameter optimization, which has the potential to greatly increase performance.

Another reason could be that the indices chosen convey somewhat redundant information, and so the neural networks' effective input feature set is smaller than it appears. The chosen indices convey only information from the R, G, and NIR bands. Perhaps the inclusion of a more diverse set of indices could have improved performance. However, *BandMdl*'s input feature set consists of data from 7 bands and performs considerably worse than *MetrMdl* with a mean-square-error of 0.2377.

The poor performance of *BandMdl* is almost certainly due to its small size.

Given larger layers and a deeper network, *BandMdl* would be able to approximate in its hidden layers features similar to the hand-engineered indices.

The main conclusion from this work is that while hand-engineered indices are a primitive way to predict physical quantities, they can perform nearly as well as small neural networks. So for machine learning performed with computational and/or time constraints, the use of indices in the input feature set offers the potential for greatly improved performance. The use of both hand-engineered features and machine learning is necessary to truly create a good predictive model in remote sensing.

Future work in forest density prediction could follow much the same approach we have used in this project but correct the errors that affect network performance. Firstly, future work should use a true in-situ groundtruth dataset. This ensures the neural networks are trained to predict the right physical quantity. It should also be ensured that there is perfect geographic alignment between the groundtruth imagery and the feature dataset. Most importantly, better neural network architectures should be implemented. This means using far more layers, using larger layers, and using an input dataset that includes raw band data and a number of vegetation indices.

Other future work could involve examining the weights of the trained network to evaluate the relative importance of input features. This would make it much easier to propose other hand-engineered features that would perform well in predicting forest density.

Finally, future work could take into account the position of pixels in relation to other pixels in training. Using a convolutional neural network either to perform segmentation as a preprocessing step or as the backbone of the network could be useful.

References

[1] https://www.gardenguides.com/13428942-how-to-calculate-tree-density.html

[2] https://land.copernicus.eu/pan-european/high-resolution-layers/forests/tree-cover-density/status-maps/2015

[3] https://scihub.copernicus.eu/dhus/#/home

[4] https://www.indexdatabase.de/db/ia.php?application_id=7