THE UNIVERSITY
of EDINBURGH

The School of Mathematics

Reconstructing Sentinel-2 Optical Data from Sentinel-1 SAR Data Using a Machine Learning Approach

by

Jonathan Hoover

Dissertation Presented for the Degree of
MSc in Statistics with Data Science

August 2022

Supervised by
Dr. Simon Taylor, Dr. Finn Lindgren, and Nina Fischer

Executive Summary

1. Introduction

Remote sensing is a critical technique used in applications like forest detection, agriculture, and disaster monitoring [1]. Applications that rely on visible light (optical data) are often impeded by cloud cover since clouds block most visible light detection. Radar, however, can pass through clouds. As such, we design an image specific machine-learning strategy to reconstruct optical data from radar data by independently predicting Red, Green, Blue, and near-Infrared (NIR) pixels given local radar data.

2. Data Source

The images in this analysis originally came from the European Space Agency's Sentinel-1 (Radar data) and Sentinel-2 (optical data) missions [2]. The images were preprocessed by the company Space Intelligence (Edinburgh) to provide geographically paired, optical images and radar images for analysis. Clouds have been removed from the optical images to provide a ground truth for comparison and testing.

3. Methods

We synthesize ten cloud regions on the Sentinel-2 data, select training and testing pixels, and transform the data into response variables and input variables suitable for a machine learning analysis. The final model takes 18 SAR variables as input to predict each of the four color bands (Red, Green, Blue, NIR). We then train a linear regressor, two decision tree regressors, and two light gradient boosting regressors and assess their reconstruction performance with both quantitative and qualitative assessments.

4. Results and Conclusions

The results suggest that the default decision tree regressor (from sklearn) consistently performs the best on assessments like residual analysis, distribution analysis, and image reconstruction. It captures the true diversity of pixel intensities for each color and can consistently produce images that are more difficult to distinguish from ground truth than the other methods. Given this model is simple to understand, easy to implement, and yields decent reconstructions, these results are a promising step towards SAR to optical image translation for cloud reconstruction.

Acknowledgments

I am grateful to the supervisors of the project, Dr. Simon Taylor, Dr. Finn Lindgren, and Nina Fischer, for their guidance. I'd also like to thank the project expert, Kristian Bodolai for his expertise.

This paper (1) Contains modified Copernicus Sentinel data 2014 for Sentinel data; and/or (2) Contains modified Copernicus Service information 2015 for Copernicus Service Information.

Modified Sentinel-1 and Sentinel-2 images were provided by Space Intelligence (Edinburgh)

University of Edinburgh – Own Work Declaration

Name: Jonathan Hoover

Matriculation Number: s2113204

Title of work: Reconstructing Sentinel-2 Optical Data from Sentinel-1 SAR Data using a Machine Learning Approach

I confirm that all this work is my own except where indicated, and that I have:

- Clearly referenced/listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Not sought or used the help of any external professional academic agencies for the work
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Complied with any other plagiarism criteria specified in the Course handbook

I understand that any false claim for this work will be penalised in accordance with the University regulations (`https://teaching.maths.ed.ac.uk/main/msc-students/msc-programmes/statistics/data-science/assessment/academic-misconduct`).

Jonathan Hoover

Date: August 9, 2022

Contents

1 Introduction

Remote sensing is a critical technique for many applications including forest detection, agriculture, and disaster monitoring [1]. These applications rely on both visible and non-visible light detection; however, many wavelengths in the visible spectrum cannot pass through clouds. Given that approximately 67% of land mass is covered by clouds, on average [3], accurate and consistent visible light measurements are difficult. Methods are needed to consistently recover visible light data.

Fortunately, radio waves, which are used in Synthetic Aperture Radar (SAR), can pass through clouds. As such, we aim to reconstruct optical data (Red, Green, Blue, and NIR) from geographically and temporally matched SAR data. We must note that radar-based remote sensing inherently captures different physical characteristics than optical data, so we expect some loss when converting between the two modalities. Nonetheless, recovering missing optical data is critical for many research applications, so we will explore the extent to which SAR data can reconstruct optical images.

In this report, we design an image specific machine-learning strategy to reconstruct Sentinel-2 optical data from Sentinel-1 SAR data by independently predicting Red, Green, Blue, and NIR pixel intensities from local SAR data (VV and VH polarization values). We then test this strategy using three types of machine learning models: Linear Regressors, Decision Tree Regressors, and Light Gradient Boosting Regressors.

The structure of this report is as follows. First, we define our data and the data manipulation methods used to create the final datasets. Then, we explain our chosen models and the methods used to assess them. Lastly, we present the reconstruction results, analyze which model produces the best reconstruction, and suggest potential avenues for future study.

2 Data Source

This study uses supervised machine learning models to map from input SAR data to output optical data. As such, we require ground truth (cloudless) images for training and testing. Ten sets of geographically paired SAR and ground truth optical images were provided by the company Space Intelligence for model training.

Images originally came from the European Space Agency's Sentinel-1 (SAR data) and Sentinel-2 (optical data) missions [2]. Space Intelligence removed clouds from the raw images to create a dataset suitable for supervised learning. Optical images were created by mosaicking several scenes of the same location (from different times) until clouds were filled. For more details, please contact Space Intelligence.

The Sentinel-1 data contains three bands of information: 1) Vertical-Vertical (VV) polarization, 2) Vertical-Horizontal (VH) polarization, and 3) Angle of sensing. Angle was discarded based on expert recommendations. Polarization is measured in decibels. The Sentinel-2 data contains 4 bands of color information: 1) Blue, 2) Green, 4) Red, 4) Near-Infrared (NIR). Values are integers ranging from 0 to 65,535

Due to computational and storage limits, only the first image (called aoi_1 in the dataset) was fully processed and analyzed. Code is available to analyze the remaining images.

3 Methods

3.1 Image Processing

3.1.1 Sentinel-1 and Sentinel-2 Image Alignment

The chosen Sentinel-2 image was in the Universal Transverse Mercator coordinate system (EPSG 32629) while the Sentinel-1 image was in the WGS 84 coordinate system (EPSG 4326). To ensure that Sentinel-1 and Sentinel-2 pixels were properly aligned, we reprojected the Sentinel-2 image into

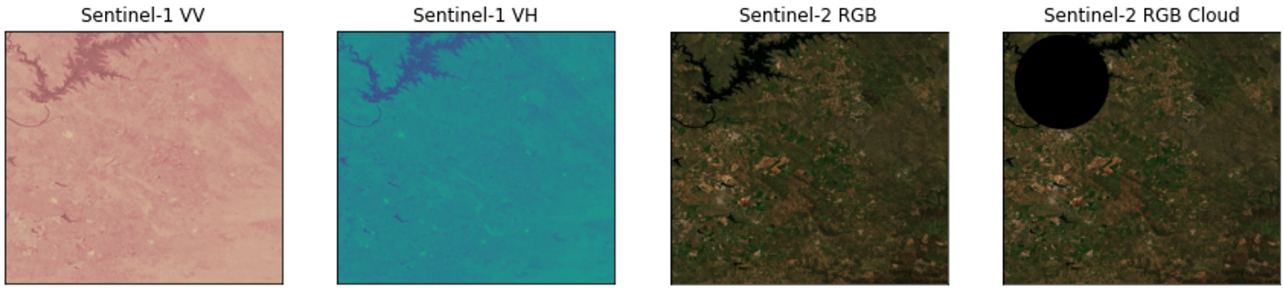


Figure 1: Fully Processed Sentinel-1 and Sentinel-2 Images. Far Left: Sentinel-1 VV Band. Middle Left: Sentinel-1 VH band. Middle Right: Sentinel-2 RGB Image. Far Right: Representative cloud image. All images were normalized to 0-1 and Sentinel-2 images were brightened for visualization.

the WGS 84 coordinated system and resampled the pixels so the resolutions matched. This processing was performed in python using the rasterio [4] package function "reproject."

After reprojection we confirmed Sentinel-1 and Sentinel-2 alignment to ensure each pixel represented the same geographic coordinates in both images. See Figure 1 for the fully processed images (only RGB is shown for Sentinel-2 for color visualization).

3.1.2 Missing Pixel Imputation

Sentinel-2 images were fully observed due to the mosaicking done by Space Intelligence. The chosen Sentinel-1 image contained one missing value, but other images in the dataset contained more missing values (although infrequent). For consistent future analysis, we chose to impute the missing value through interpolation using the "fillnodata" function in rasterio. Since missingness was infrequent, interpolation should not introduce much bias.

3.1.3 Image Cropping

Future analyses will explore deep learning models that require small image tiles (256 by 256 pixels) as input. To mainstream future analysis, we cropped the image so the final dimensions allow for easy 256 by 256 pixel tiling. Cropped images are shown in Figure 1).

3.1.4 Cloud Synthesis

The goal of this study is to develop models that reconstruct missing optical pixels (due to cloud coverage) from SAR data. To simulate image reconstruction we defined cloud pixels that could be used to test the reconstruction process.

Traditionally, a test set would simply contain a random sample of the dataset. However, clouds are often contiguous; to capture this structural continuity, we generated random circular regions in the image and selected pixels within this region for the cloud (test) set. We chose to generate circles covering approximately ten percent of the image and ensured that the regions do not overlap with edges.

We repeated this process ten times to generate ten random cloud sets. More samples should be generated to improve robustness, but ten is sufficient for the scope of this report. Similarly, the effect of cloud size should assessed, but we do not consider cloud size for this analysis. Representative images of Sentinel-1, Sentinel-2, and a cloud image are shown in Figure 1.

3.2 Feature Engineering

3.2.1 Generating Response Variables

Traditional supervised learning models require a feature matrix ($\mathbf{X}$) and response vector ($\mathbf{y}$). We will develop a unique model for each Sentinel-2 color band (Red, Green, Blue, NIR) since the mapping

between SAR and each color is likely unique. As such, we need four response vectors corresponding to unique color intensities for training and testing.

We generated the four test set response vectors by selecting the pixels (for each band) from the previously defined cloud pixels. With a ten percent cloud size, this yielded four vectors of size 720,437. This corresponds to 9.999% of the total pixels (per band).

We generated the training set response vectors by randomly selecting non-cloud pixels from each band (The same pixel indices are used for each color). We chose a sample size of 1,000,000 for this analysis. Using all non-cloud pixels (6,488,523 for this image) would yield better results, but we subset due to computational and memory limits. This training size covers approximately 13.87% of the data.

This selection process was performed for each of the ten cloud images generated. The resulting training and test sets are all the same size.

3.2.2 Generating Feature Matrices

We will use SAR data to reconstruct optical data, so the feature matrix ($\mathbf{X}$) must contain some combination of SAR VV and VH band data. For each response pixel (y_i) we will use the corresponding SAR pixel from VV and VH along with the pixels that surround this pixel as shown in Figure 2. Additionally, we will create separate variables for VV/VH values to see if there is any interaction. Thus, the initial feature matrix will contain 27 variables (9 for VV, VH, and VV/VH, respectively).

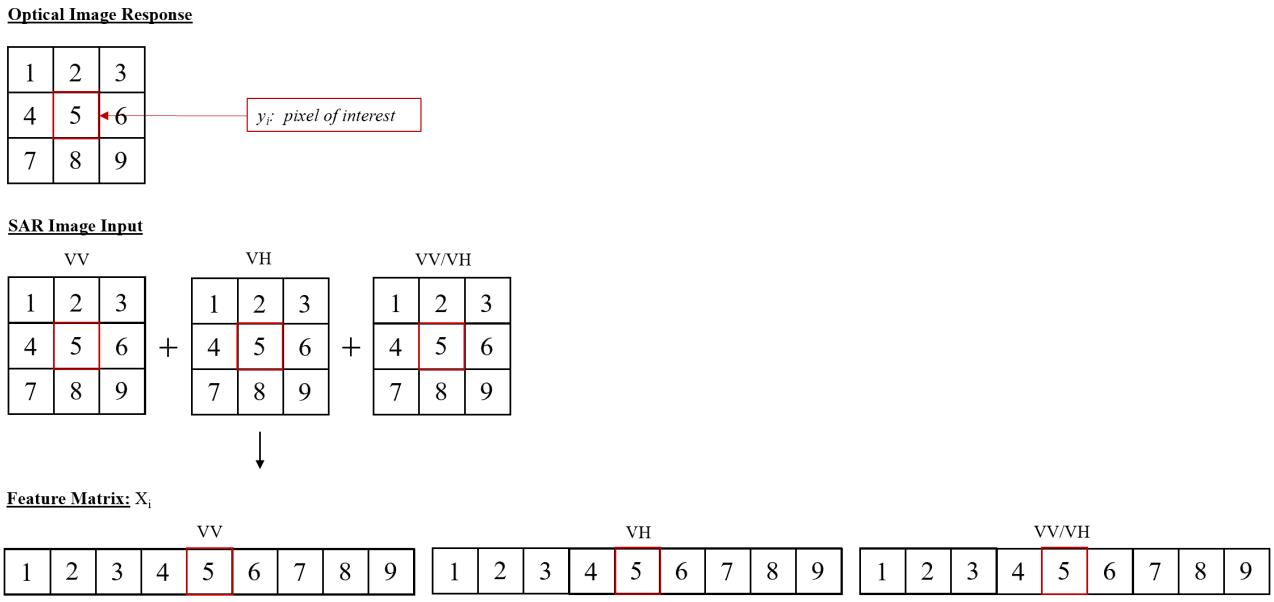


Figure 2: Initial Feature matrix (X) design process. The pixels in Sentinel-1 surrounding the pixel of interest (5 in the Image) are taken as input features for VV, VH, and VV/VH

3.2.3 Feature Selection

We explored the relationship between the initial input features and the response variables to determine if any variables should be dropped. Figure 3 shows the relationships between the central SAR pixel (Pixel 5 for VV , VH, and VV/VH in Figure 2) and the color response variables. Figure 4 shows the relationship between the bottom right pixel (Pixel 9 in Figure 2) and the color response variables.

Figures 3 and 4 demonstrate a clear relationship between the VV and VH intensities and color intensity. There is no clear relationship between VV/VH and the color data, so VV/VH features will be dropped. Note that we also examined the relationship between the remaining input variables (Pixels 1-4 and 6-8 in Figure 2) and color intensities; the observed trends hold for the remaining variables (not shown for brevity). The final feature matrix has 18 feature variables (9 for each of VV and VH).

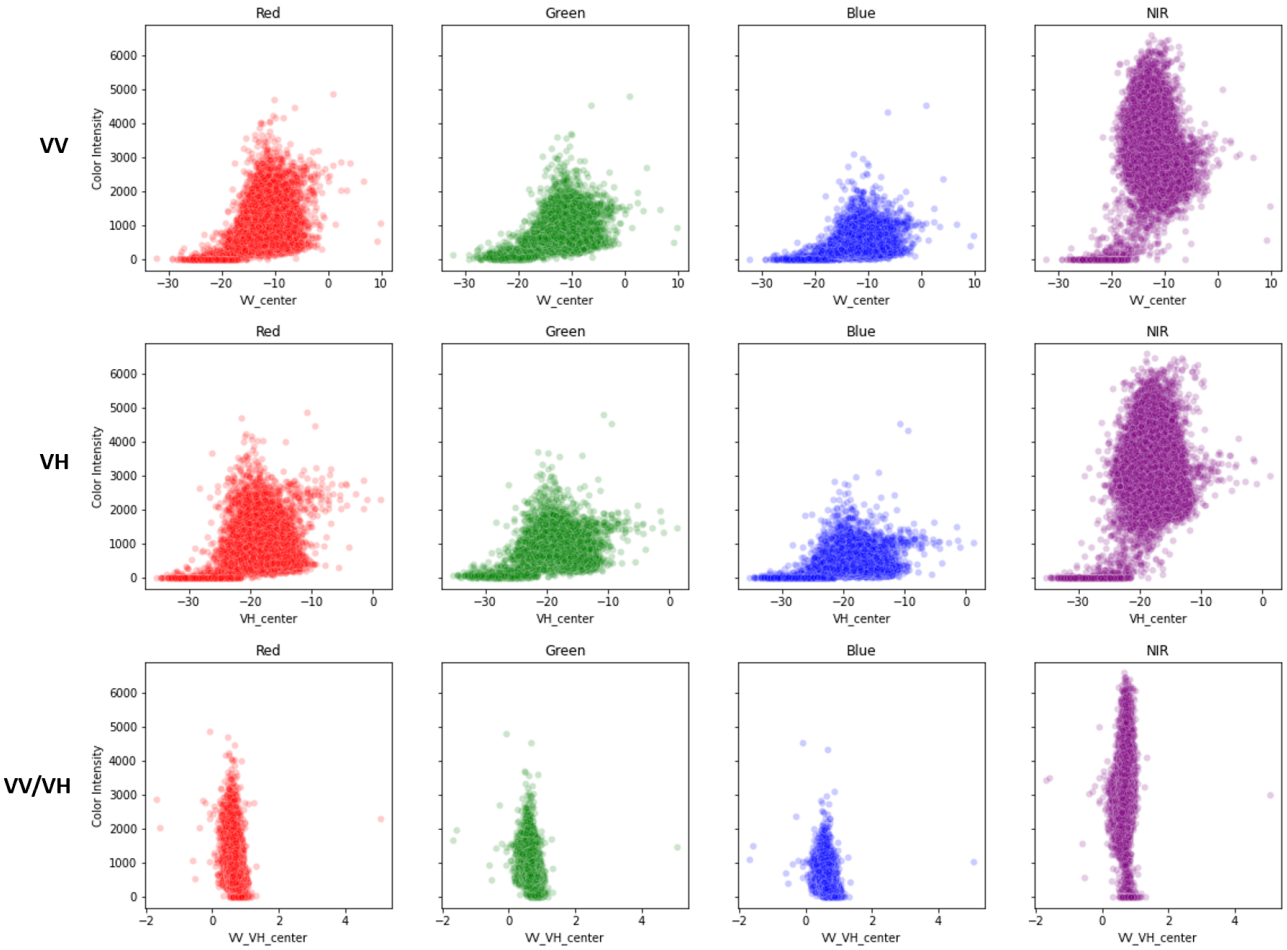


Figure 3: Relationship between Sentinel-1 central pixel intensities and Sentinel-2 color intensities. Here the Sentinel-1 pixel is the central pixel (Pixel 5 in Fig 2). Top: VV vs Colors. Middle: VH vs Colors. Bottom: VV/VH vs Colors

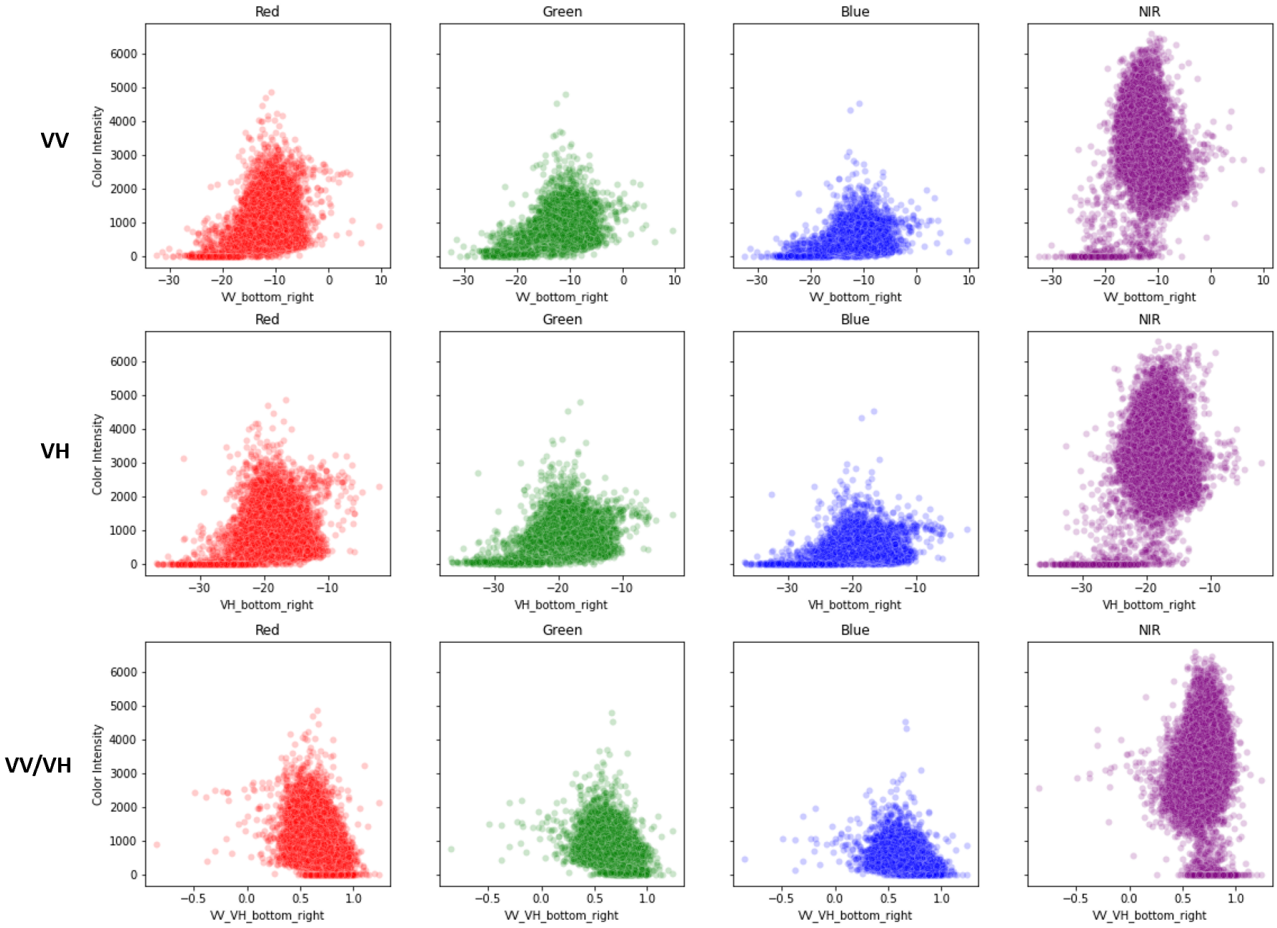


Figure 4: Representative relationship between Sentinel-1 off-center pixel intensities and Sentinel-2 color intensities. Here the Sentinel-1 pixel is the bottom right (Pixel 9 in Fig 2). Top: VV vs Colors. Middle: VH vs Colors. Bottom: VV/VH vs Colors

3.3 Models

To identify potential models for analysis, we explored the relationship between the SAR input and the optical output further.

Figures 3 and 4 suggest that SAR and optical color data are not linearly related. Instead, there appears to be a bimodal relationship between the SAR variables and the optical outputs. For VV, values below (approximately) -18 seem to fall in one category while values above fall into another. VH shows a similar pattern, but the splitting threshold appears to be (approximately) -28.

It is possible that these distinct groups describe different landscapes, so we created a binary mask for the images to check if the groups correspond to unique geographical traits. If a pixel's intensity is below the threshold, we set the mask equal to 1 and if it is above the threshold we set the mask equal to 0. This was done for both VV and VH.

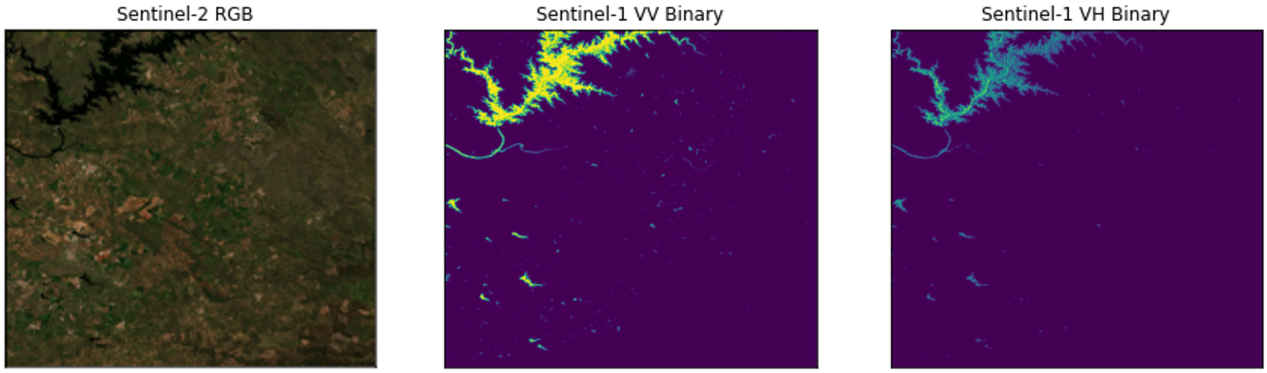


Figure 5: The relationship between Sentinel-1 and Sentinel-2 depends on the type of geography present. Left: Sentinel-2 RGB. Middle: Sentinel-1 VV Binary Mask; Colorized parts represent VV values below -18. Right: Sentinel-1 VH Binary Mask; colorized parts represent VH values below -28. Original Sentinel-2 image was too dark to easily view, so brightness, exposure, and contrast were edited in Microsoft Photos for visualization.

Figure 5 shows the plot of the binary masks for VV and VH alongside the original Sentinel-2 (RGB) Image. The pixels below the thresholds (-18 and -28 for VV and VH, respectively) appear to belong to a river. These results suggest that the mapping between SAR and color data may depend on the geographical structures under assessment (water in this case).

As such, we need models that can handle both nonlinear data and data with mixed SAR to optical relationships (data is numerical, so we require a regressor). This data structure immediately suggests two model types: Decision tree regressors and deep learning neural networks. For this study we focus on decision tree based regressors, but future analysis should explore deep learning networks, particularly conditional Generative Adversarial Networks, which have been shown to work for this type of problem [5].

Since the data appears nonlinear, we will use the linear model as a baseline that we can improve upon.

3.3.1 Linear Regression Model

Linear models are the classic regression model. They assume a linear relationship between a continuous response variable y and a set of input variables $x_1, ..., x_p$ such that:

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon \tag{3.1}$$

Where $\mathbf{Y}$ is an (N x 1) column vector of responses, $\mathbf{X}$ is an (N x (p+1)) feature matrix, β is a ((p+1) x 1) column vector of coefficients, and ϵ is an (N x 1) column vector of residuals (this equation assumes an intercept). Additionally, the models assume that 1) $y_i \sim N(\mu, \sigma^2)$ where μ is the population mean

and σ^2 is the (constant) population variance and 2) the residuals, e_i, are independent and identically distributed $N(0, \sigma^2)$.

These models are quite useful for normally distributed, linear data. However, since our data does not appear to be linear, we do not expect the model to perform well. As such, we use this model as a baseline for comparison with the other models.

3.3.2 Decision Tree Regression Model

Decision trees are a non-parametric technique used to partition data based on internal data characteristics [8]. The partitioning creates distinct categories (called decision nodes) by finding the splitting criteria that maximizes homogeneity of the resulting nodes. These boundaries can then be used to predict the categories of unseen data. This technique can also be applied in a regression setting.

Regression trees generate numerical predictions by passing input variables to the tree, identifying the decision node, and assigning the average response value (from training observations) associated with the decision node. Importantly, they are nonlinear and can handle mixed relationships between input and output data, so they are a good fit for our data.

We must note that decision trees are prone to overfitting since they can generate incredibly complex tree structures. However, our reconstruction method depends on building a unique model for each image to capture its geographic signature. Thus, overfitting to an image is our goal and these models are valuable to consider. Nonetheless, if an image has widely varying geographies, local overfitting may still be a concern. We will assess this by testing on random cloud locations.

We implement the DecisionTreeRegressor in python's scikit-Learn package.

3.3.3 Light Gradient Boosting Regression Model

The Light Gradient Boosting (LGB) regressor is an ensemble decision tree regressor. As an ensemble method, it combines the results of a group of weak decision trees to produce a more robust result than a single tree would alone (by reducing variance [7]). As a gradient boosting method, it trains individual trees in sequence, calculates prediction errors, and minimizes the error from the previous tree to improve the results of the subsequent tree [9] [6].

The LGB regressor is a "light" gradient booster because it attempts to reduce the training size and feature space without reducing information content. The model randomly drops data points with a small gradient (low impact on fit) and combines sparse features (features with many zeros) into one. This retains critical gradient and feature information while reducing the training size. For more details see [6].

Because it effectively reduces the training size without losing information, the LGB regressor is efficient and accurate for large datasets (like our image data). Additionally, it is inherently nonlinear and can handle mixed relationships in the data. As such, it is a valid modelling approach for our data. Again, this type of model is prone to overfitting due to the tree-based nature, but, as mentioned before, overfitting could be useful for our purposes. We implement the LGBMRegressor in python's lightgbm package.

We must note that the random forest regressor and extreme gradient boosting (XGBoost) regressor would be worth trying on this data, but due to time restraints we limit our analysis to the models specified here.

3.3.4 Model Tuning

Decision tree models contain hyperparameters that require tuning for optimal performance. Due to time and computational restraints, we only tuned on a 10,000 sample subset of the training data. Additionally, we used a random search instead of a grid search to explore the parameter space. 10 random parameter combinations were searched and assessed using the default scoring parameter. Each

color regressor was tuned separately. For more robust and representative results, tuning should be performed with a larger sample size using a grid search.

For the decision tree regressor, we tuned the max feature and max depth parameters. Max features defines how many features to consider for node splitting and max depth defines how deep the tree can grow.

The optimal hyperparameters for each color regressor were: 1) Red: max depth = 5, max features = 'sqrt'; 2) Green: max depth = 5, max features = 'sqrt'; 3) Blue: max depth = 5 and max features = None; 4) NIR: max depth = 5 and max features = 'sqrt.'

For the LGB regressor we tuned the number of estimators, max depth, learning rate, and minimum child weight parameters. Max estimators defines the maximum number of sequential trees to build, learning rate defines the magnitude of sequential tree modification (lower is slower and more precise), and minimum child weight limits tree depth.

The optimal parameters for each color regressor were: 1) Red: number of estimators = 100, learning rate = 0.05, max depth = 20, and minimum child weight = 4; 2) Green: number of estimators = 100, max depth = 15, learning rate = 0.05, and minimum child weight = 0.001; 3) Blue: number of estimators = 100, max depth = 5, learning rate = 0.05, and minimum child weight = 2; 4) NIR: number of estimators = 100, max depth = 20, learning rate = 0.1, and minimum child weight = 4.

Note that we also run the default parameters for the decision tree and LGB regressors as a comparison.

3.3.5 Model Assessment Strategies

We will use traditional statistics like root mean squared error (RMSE: $\sqrt{\frac{\sum_{i=1}^{N} y_i - \hat{y_i}}{N}}$) and mean absolute percentage error (MAPE: $\frac{1}{N}|\frac{\sum_{i=1}^{N} y_i - \hat{y_i}}{y_i}|$) to provide quantitative estimates for the average error of a model. However, because these are point estimates for average error, we do not expect them to capture the full structural complexity of a diverse image dataset. Nonetheless, they will let us quantitatively assess how consistent models are across all cloud samples.

We will also use more qualitative assessments like distribution analysis and visual image reconstruction. A distribution analysis will let us compare the true and predicted pixel distributions to identify how well the model captures the true diversity present in an image (for instance bimodal pixel distributions should be captured). Viewing the predicted image will help us assess how "real" a reconstructed image appears. If a human cannot distinguish between the reconstructed and ground truth image, the reconstruction method works well.

Combining traditional metrics like RMSE and MAPE with more nuanced assessments like the distribution analysis and visual inspection will let us holistically assess how well a model performs.

4 Results

4.1 Residuals Analysis

Figure 6 shows the true versus predicted Sentinel-2 cloud pixel values. The default decision tree regressor (second from top) appears to perform the best since it is the only model where the points are randomly scattered about the line $y = x$ for all colors. The LGB models (default: second from bottom; tuned: bottom) both appear to predict red, green, and blue well, but they underpredict NIR for large values. The tuned decision tree (middle) appears to predict discrete values, suggesting the tree was not deep enough to capture the complexity of the data. Lastly, the linear model (top) appears to predict essentially the same value regardless of true value. Overall, the errors are widely distributed for each model, but the default decision tree appears to yield the most consistent predictions.

Interestingly, the tuned models appear no better (and occasionally worse) than the default models. This is likely because we only performed a random search on 10,000 samples from the training data.

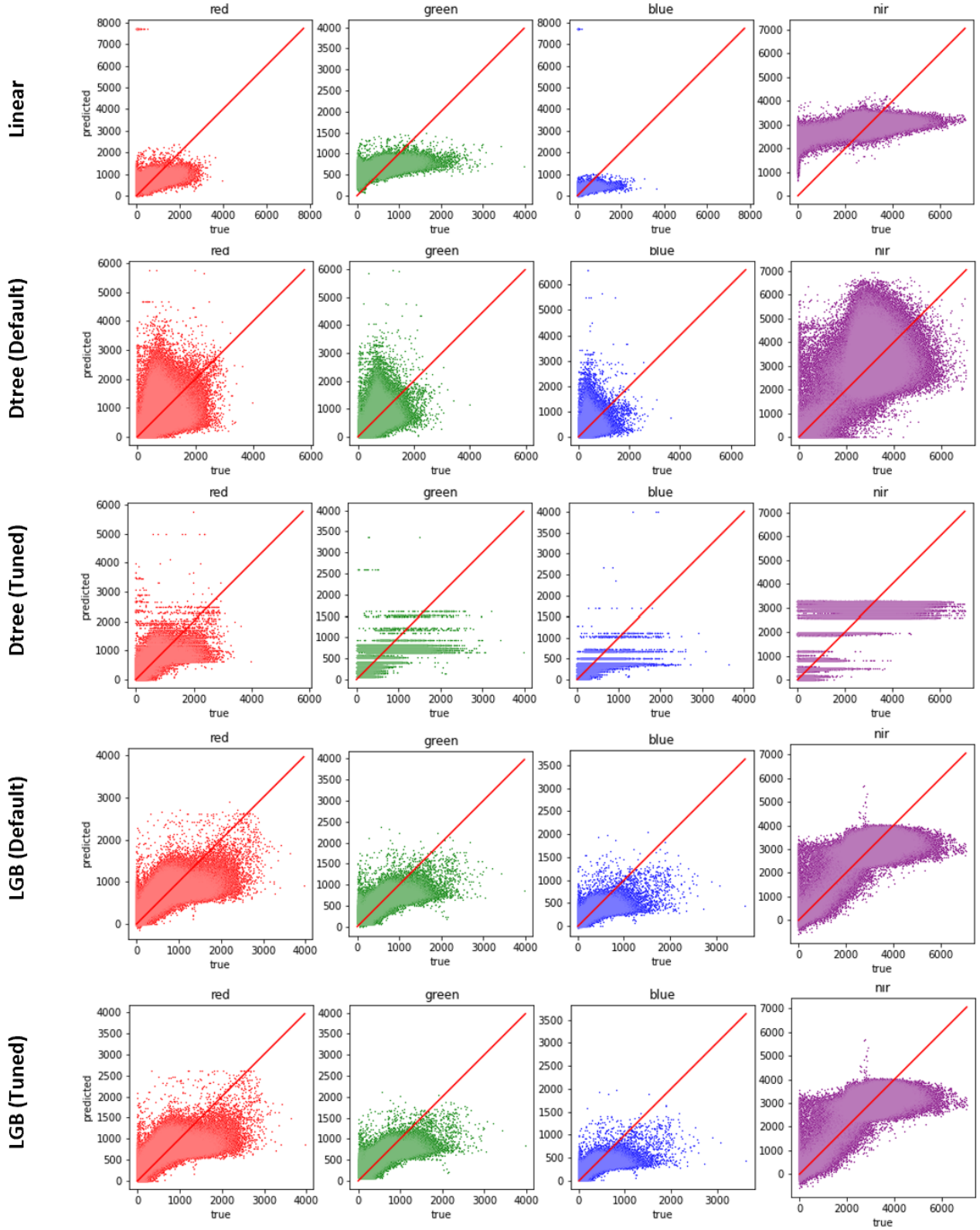


Figure 6: True vs. predicted Sentinel-2 cloud pixel intensities for cloud 1. Models from top to bottom: Linear, Decision Tree (Default), Decision Tree (Tuned), LGB Regressor (Default), LGB Regressor (Tuned). The line for $y = x$ is shown in red.

For better results, a grid search should be done on a larger subset of data.

We analyzed the same plots for each of the 10 cloud images, and the results were consistent with those shown here.

4.2 Distribution Analysis

Figure 7 shows the true versus predicted cloud pixel distributions for each model (for one representative cloud image). The true data shows a bimodal distribution for each Sentinel-2 color band. In Figures 3, 4, and 5 we identified this bimodal distribution and found that one of the distributions was associated with river pixels (see Figure 5. Figures 3, 4 and 5 suggest that the river is associated with low intensity Sentinel-2 pixel values, so the left peak in Figure 7 likely corresponds with river pixels. Thus, the right peak is likely associated with land pixels.

Overall, the default decision tree regressor (second from the top) appears to capture the pixel distribution best for all colors. While its predicted distribution is slightly different than the truth, it captures the true data's bimodal structure and diversity well (the peaks are only slightly wider than the true peaks). The LGB models (default: second from bottom; tuned: bottom) capture the river peak well; however, they appear to lack diversity in land predictions (their land peaks are much narrower than the true distribution). The tuned decision tree (middle) again appears to predict discrete values (there are numerous peaks), suggesting the model is underfit. The linear model (top) again performs the worst since it is unable to capture the bimodal distribution and clusters around the average for the land peak. Note that most models appear unbiased since they cluster around the mean values for each peak, yet only the default decision tree appears to capture the diversity and kurtosis of the true data.

We looked at the distributions for each of the 10 cloud images (not shown for brevity), and the results were consistent with those shown here: the default decision tree captures the distributions best.

4.3 Model Consistency Analysis

Our reconstruction strategy requires fitting a model to every new image for reconstruction. As such, we expect models to overfit to an individual image. However, we do not want models to be dependent on the location of a cloud within an image. They should be robust enough to handle local geographic variations present within an image. To assess how robust our reconstruction models are to variation in cloud location, we tested the models on ten randomly generated models. We report the Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and reconstructed image for each sample below.

Note that the previous analyses (residuals and distribution analyses) also compared results across the 10 cloud samples and found consistent graphical results; however, we do not show each plot for brevity.

4.3.1 RMSE and MAPE

Figure 8a) shows the mean RMSE with 95% confidence intervals for each model across 10 cloud sampling runs. The results suggest that RMSE (Figure 8a) depends on color. Ranking the colors based on model performance gives the order blue, green, red, and NIR where blue is the color that models predict the best.

Results within colors are mostly consistent. LGB models yield the lowest RMSE (tuned and default perform about the same) for all colors. Since LGB models are optimized for gradients and ensemble models perform better than single models, these results are expected. The tuned decision tree has the next lowest RMSE for all colors, which makes sense since tuning optimizes for average error. The linear model and default decision tree are essentially tied for last; the default decision tree performs worse for green, better for blue, and approximately the same for red and NIR.

Overall, most models produce narrow confidence intervals (except the linear model) for RMSE, suggesting that average error is consistent across all cloud samples (See Supplementary Table 1 for all 95% confidence intervals). The confidence intervals are wider for the NIR band than the remaining

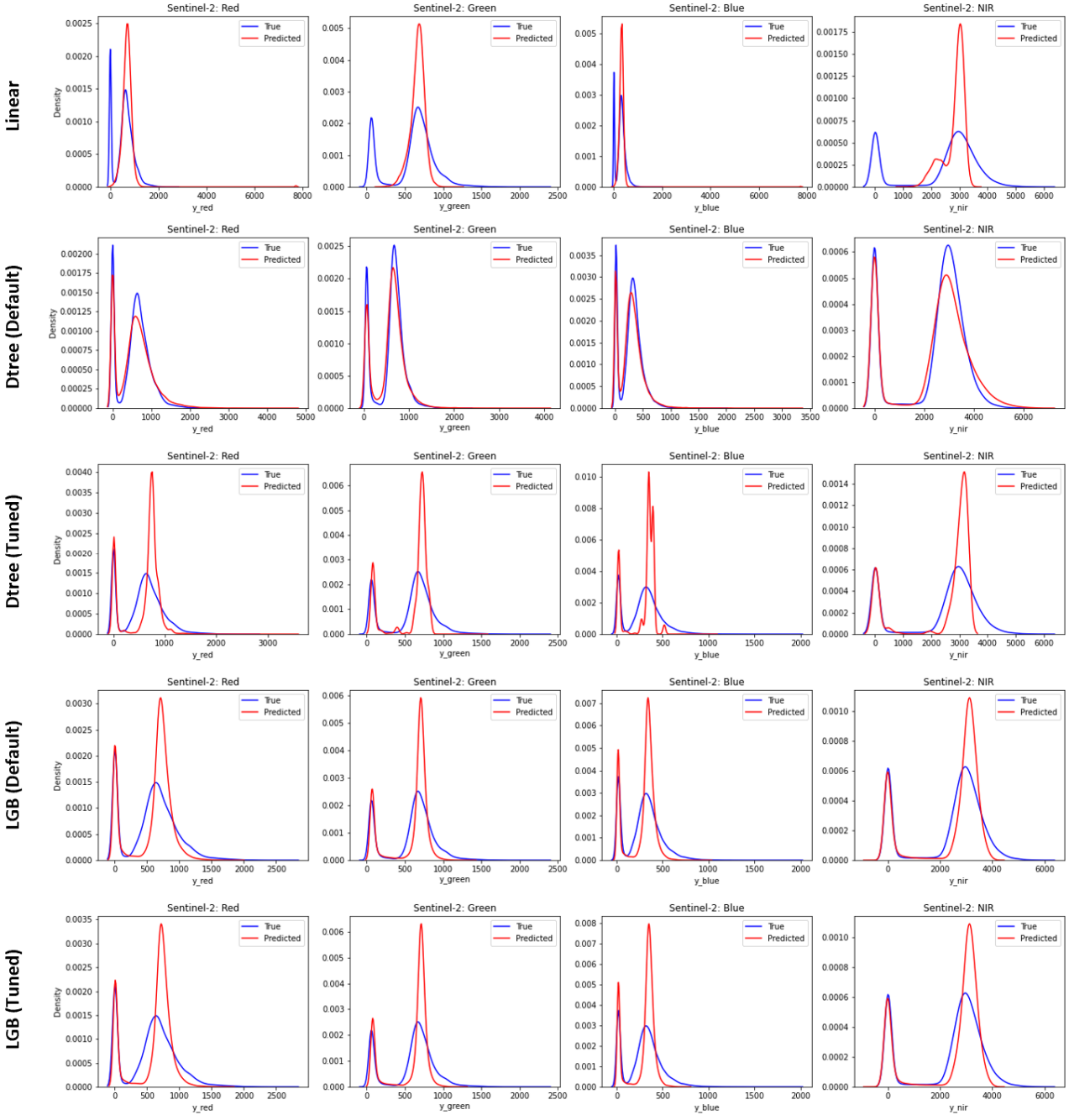
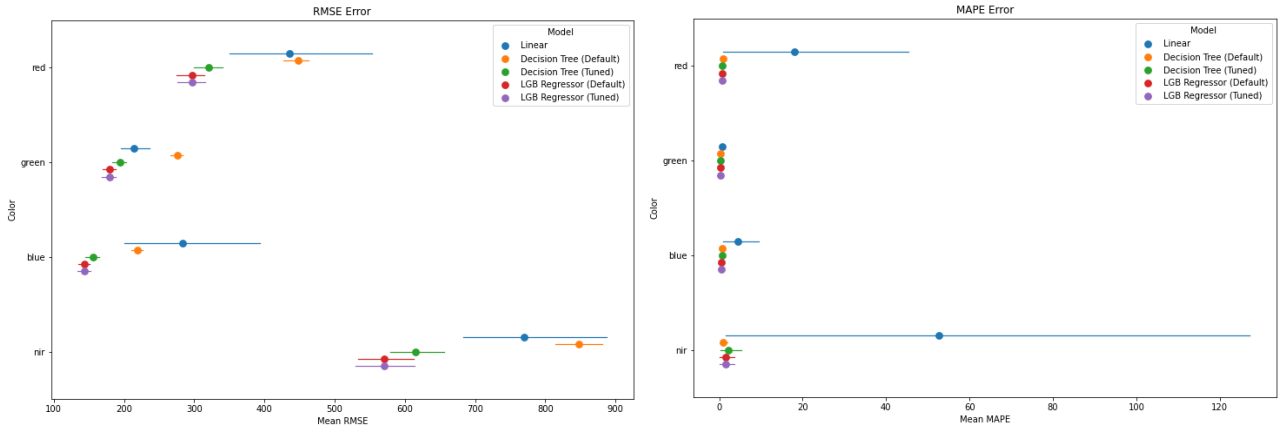


Figure 7: Distribution of true and predicted Sentinel-2 cloud pixel intensities for cloud 1. Models from top to bottom: Linear, Decision Tree (Default), Decision Tree (Tuned), LGB Regressor (Default), LGB Regressor (Tuned). True is shown in blue and predicted is shown in red. The distributions were plotted using a normal kernel density estimator on 100,000 samples (true/predicted matched) from the cloud testing dataset instead of the full dataset for computational efficiency.



(a) RMSE **(b)** MAPE

Figure 8: Mean RMSE and MAPE statistics with 95% confidence intervals for each model. Averages were calculated for RMSE and MAPE across the 10 cloud samples. Confidence intervals were calculated via a 1000 sample bootstrap.

colors, suggesting that SAR struggles to consistently reconstruct NIR compared to the other colors. The wide interval for the linear model is likely due to poor performance for water regions as discussed in section 4.2).

Figure 8b) shows the mean MAPE with 95% confidence intervals for each model across 10 cloud sampling runs. MAPE is consistently low for all models except for the linear model in the red and NIR bands (again likely due to its poor performance on water regions).

MAPE calculates error relative to the true value, so low MAPE values indicate the error is small relative to the true value and high values indicate the error is large relative to the true value. MAPE can be difficult to interpret when the scale is close to zero (dividing by small numbers yields large MAPEs), but Figure 6 shows the residuals are quite large. As such, low MAPE values for most models suggest that errors are small relative to true pixel values for all colors.

RMSE and MAPE both measure average error. However, simply approaching the true value on average does not necessarily mean a good reconstruction. Models that perform well with RMSE and MAPE (LGB regressors and tuned decision tree) may produce smoothed reconstructions that do not capture the complexity of an image. Additionally, models that perform poorly with them might capture the structural complexity. For instance, the default decision tree regressor performs the best for distribution analysis, yet it performs almost the worst for RMSE. These results suggest there is a tradeoff between recovering structural complexity and minimizing average error. Nonetheless, these results suggest that model performance is invariant to cloud location (except for the linear model).

4.3.2 Image Reconstruction

Residuals, pixel distributions, RMSE, and MAPE are useful because they can quantify distinct aspects of a model's prediction capacity, but they cannot capture every detail. The best way to see how close a reconstruction is to the truth is to simply view the image. We may not be able to quantify our visual inspection, but it is a useful tool for quickly identifying models that cannot reconstruct an image.

Figure 9 shows reconstructed Sentinel-2 RGB images for each model.

The LGB regressors clearly struggle to reconstruct the data visually. The default LGB model can only predict black images. The model is likely predicting a high intensity outlier somewhere that causes the remaining pixels to look black after normalization, but we could not confirm this in the data. The tuned LGB model can visualize nine out of the 10 clouds, but it still yields one black image. For those that it can visualize, the cloud regions are visually noticeable after close inspection, but they appear to reconstruct some of the structural components of the ground truth image. This suggests that further

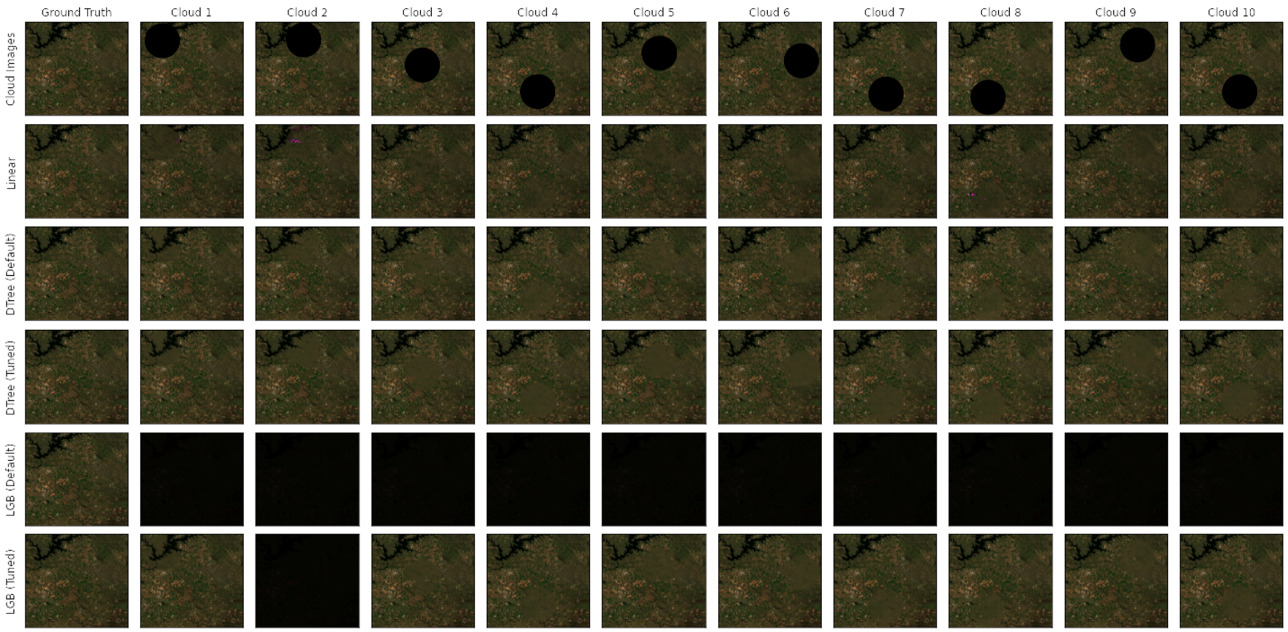


Figure 9: Cloudy Sentinel-2 RGB images reconstructed from Sentinel-1 SAR Data. Panels from top to bottom: Cloudy images, Linear model reconstruction, Default Decision Tree reconstruction, Tuned Decision Tree reconstruction, Default LGB reconstruction, and Tuned LGB reconstruction. Original images were too dark to easily view, so brightness, exposure, and contrast were edited in Microsoft Photos for each model image (consistently across all models) for visualization.

tuning (particularly with a grid search and more samples) may improve model consistency.

The tuned decision tree regressor appears to smooth the color predictions over the cloud region to produce a green blur, which makes the cloud regions immediately noticeable. This is likely because the tuned decision tree only predicts discrete pixel outcomes, so it cannot capture any of the complexity in the image.

The linear model performs surprisingly well; some of the cloud regions are only noticeable after close inspection. Nonetheless, it still struggles to reconstruct the river and occasionally produces pink dots in the cloud regions (likely due to water-coverage), so it is not a consistent model.

The default decision tree appears to produce the best reconstructed images. Some of the cloud reconstructions are not immediately detectable (particularly cloud 2) and it appears to capture some structural complexity in the forest. After closer inspection, it still lacks precise granularity, but the reconstructions are consistently difficult to distinguish from ground truth (relative to the other models).

Image reconstruction lets us assess how well a model can produce a "realistic" image. Additionally, it lets us assess how well the models perform on some of the noticeable geographic characteristics like the river in our image. But most importantly, it immediately identifies poor visual reconstructions (default LGB and tuned decision tree). Nonetheless, we cannot quantify differences, so we must combine visual inspection with other metrics to assess the model holistically. Combining the image reconstruction with RMSE and MAPE results suggests that the default decision tree regressor is the best at consistently constructing images that are difficult to distinguish from ground truth (relative to the other models).

Note that false color images were also produced to check reconstruction of the NIR channel. The results were consistent with those for the RGB channel (default decision tree was the best), but the images are not presented here for brevity (See Supplementary Figure 10 for the NIR shifted image).

5 Conclusions

Overall, we designed an image specific machine-learning strategy to reconstruct Sentinel-2 optical data from Sentinel-1 SAR data by independently predicting Red, Green, Blue, and NIR pixel intensities given local VV and VH polarization values. We then trained a linear regressor, two decision tree regressors, and two light gradient boosting regressors and assessed the modelling approaches with both quantitative and qualitative assessments.

The results suggest that the default decision tree regressor (from sklearn) consistently performs the best on qualitative assessments like residual analysis, distribution analysis, and image reconstruction. The distribution analysis suggests that it captures the true diversity of pixel intensities and the image reconstruction demonstrates that it can consistently produce images that are more difficult to distinguish from ground truth than the other methods. Additionally, while it yielded some of the highest RMSE values, the confidence intervals were narrow, emphasizing that it is invariant to cloud location.

In the end, none of the reconstructions were completely undetectable to the human eye, but the default decision tree regressor was the best of those tested because it produces images that match the distribution of the true result and require close inspection to distinguish from ground truth. Given this model is simple to understand, easy to implement, and yields decent reconstructions, these results are a promising step towards SAR to optical image translation for cloud reconstruction.

5.1 Limitations

While this analysis provides useful techniques for cloud reconstructions, there are important limitations. The first, and most important, is that this analysis was performed only on one image due to time and computational restraints, so we cannot be certain that our process would produce similar results on another image. This must be confirmed in future analyses. Second, for the image we processed, we subset both the training (n = 1,000,000) and hypertuning (n = 10,000) data for computational reasons. Future analyses should use more of the available data and should use a grid search for hypertuning instead of a random search. Lastly, we only considered models that reconstruct the whole image. Better results may be achieved for models that attempt to reconstruct an individual output such as vegetation index (particularly the NDVI index).

5.2 Future Directions

Our analysis found that distribution analysis results closely matched image reconstruction results. As such, it may be useful to define a loss function that minimizes overall distribution loss rather than average loss. Kullback-Leibler Divergence [11] is a potential metric since it measures the difference between two probability distributions.

Lastly, much of the active research on SAR to optical image translation seems to converge on deep learning models called conditional Generative Adversarial Networks [5] [10]. A preliminary attempt at the model proposed in [5] yielded results that could easily be improved with more time (Supplementary Figure 11. Future analysis should focus on tuning and improving these types of models.

References

[1] Ranganath R. Navalgund, V. Jayaraman, and P. S. Roy. Remote sensing applications: An overview. *Current Science*, 93(12):1747–1766, 2007. ISSN 00113891. URL `http://www.jstor.org/stable/24102069`.

[2] European Space Agency. Copernicus sentinel data 2014,2015, 2014-2015.

[3] Michael D. King, Steven Platnick, W. Paul Menzel, Steven A. Ackerman, and Paul A. Hubanks. Spatial and temporal distribution of clouds observed by modis onboard the terra and aqua satellites. *IEEE Transactions on Geoscience and Remote Sensing*, 51:3826–3852, 7 2013. ISSN 0196-2892. doi: 10.1109/TGRS.2012.2227333.

[4] Sean Gillies et al. Rasterio: geospatial raster i/o for Python programmers, 2013–. URL `https://github.com/rasterio/rasterio`.

[5] Lei Wang, Xin Xu, Yue Yu, Rui Yang, Rong Gui, Zhaozhuo Xu, and Fangling Pu. Sar-to-optical image translation using supervised cycle-consistent adversarial networks. *IEEE Access*, 7: 129136–129149, 2019. ISSN 2169-3536. doi: 10.1109/ACCESS.2019.2939649.

[6] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. 30, 2017. URL `https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf`.

[7] Clifton D. Sutton. *Classification and Regression Trees, Bagging, and Boosting*. 2005. doi: 10.1016/S0169-7161(04)24011-1.

[8] Wei-Yin Loh. Classification and regression trees. *WIREs Data Mining and Knowledge Discovery*, 1:14–23, 1 2011. ISSN 1942-4787. doi: 10.1002/widm.8.

[9] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001. ISSN 00905364. URL `http://www.jstor.org/stable/2699986`.

[10] Mario Fuentes Reyes, Stefan Auer, Nina Merkle, Corentin Henry, and Michael Schmitt. Sar-to-optical image translation based on conditional generative adversarial networks—optimization, opportunities and limits. *Remote Sensing*, 11:2067, 9 2019. ISSN 2072-4292. doi: 10.3390/rs11172067.

[11] Taehyeon Kim, Jaehoon Oh, NakYil Kim, Sangwook Cho, and Se-Young Yun. Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation. 5 2021.

Supplementary Tables and Figures

Model	Color	Mean RMSE	RMSE LB	RMSE UB	Mean MAPE	MAPE LB	MAPE UB
Decision Tree (Default)	blue	219.424	211.215	226.092	0.721	0.503	1.14
Decision Tree (Default)	green	275.404	266.248	282.989	0.337	0.296	0.41
Decision Tree (Default)	ndvi	0.227	0.199	0.265	2.98e+13	4.31e+11	7.05e+13
Decision Tree (Default)	nir	847.343	814.188	882.622	0.911	0.245	2.07
Decision Tree (Default)	red	447.872	425.253	463.303	0.869	0.5	1.5
Decision Tree (Tuned)	blue	155.885	145.365	165.12	0.65	0.384	1.17
Decision Tree (Tuned)	green	193.918	183.272	203.305	0.262	0.214	0.346
Decision Tree (Tuned)	ndvi	0.173	0.148	0.204	1.9e+13	3.26e+11	4.49e+13
Decision Tree (Tuned)	nir	615.167	578.503	656.74	2.21	0.202	5.76
Decision Tree (Tuned)	red	320.642	296.405	340.717	0.773	0.375	1.43
LGB Regressor (Default)	blue	143.153	134.774	151.692	0.554	0.353	0.943
LGB Regressor (Default)	green	179.017	168.769	187.797	0.237	0.201	0.301
LGB Regressor (Default)	ndvi	1.62	0.246	3.914	1.12e+14	1.16e+12	3.05e+14
LGB Regressor (Default)	nir	570.935	529.319	611.492	1.56	0.183	3.62
LGB Regressor (Default)	red	296.441	275.156	313.875	0.7	0.352	1.23
LGB Regressor (Tuned)	blue	143.818	135.173	152.377	0.575	0.36	0.993
LGB Regressor (Tuned)	green	179.074	169.239	188.028	0.242	0.201	0.317
LGB Regressor (Tuned)	ndvi	2.336	0.82	4.359	1.83e+14	4.62e+12	4.58e+14
LGB Regressor (Tuned)	nir	570.935	532.22	612.133	1.56	0.182	3.62
LGB Regressor (Tuned)	red	297.314	275.825	315.427	0.772	0.357	1.43
Linear	blue	283.444	203.92	395.731	4.39	1.05	9.19
Linear	green	214.354	195.875	236.525	0.651	0.256	1.35
Linear	ndvi	0.407	0.231	0.635	7.3e+13	1.65e+12	1.69e+14
Linear	nir	769.857	676.75	891.087	52.8	1.38	140
Linear	red	435.878	353.57	557.965	18.1	0.885	42

Table 1: 95% Confidence intervals for mean RMSE and MAPE across 10 cloud samples.

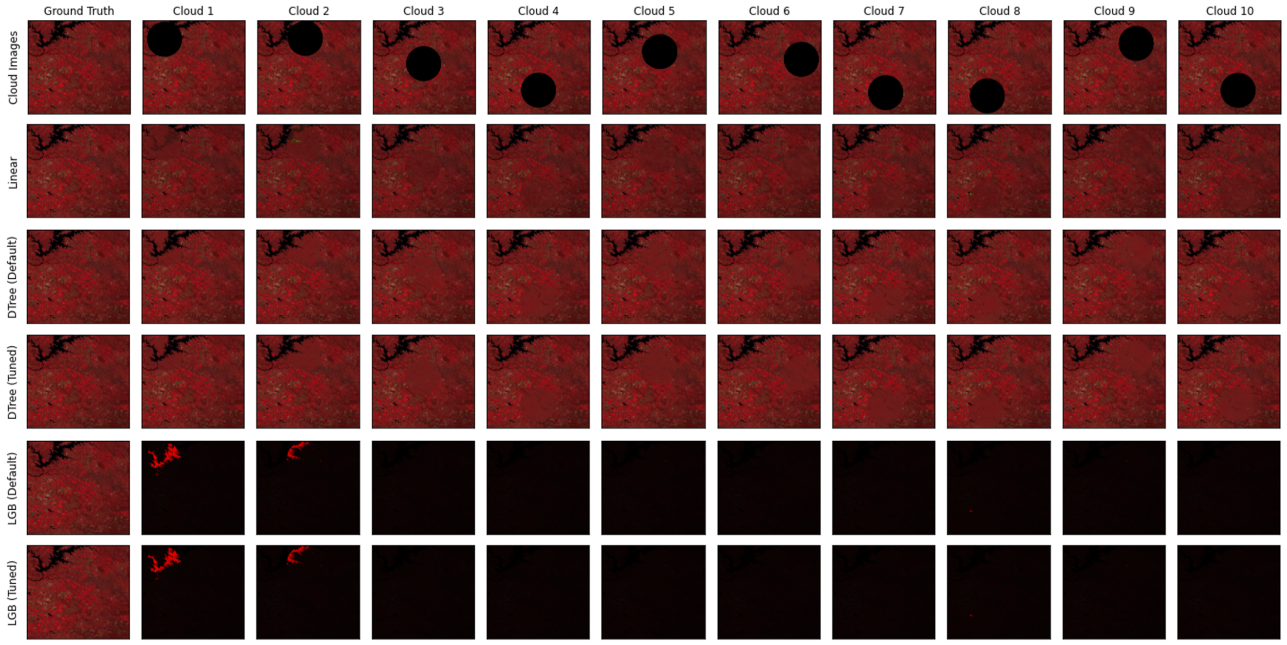


Figure 10: Cloudy Sentinel-2 NIR shifted images reconstructed from Sentinel-1 SAR Data. Panels from top to bottom: Cloudy images, Linear model reconstruction, Default Decision Tree reconstruction, Tuned Decision Tree reconstruction, Default LGB reconstruction, and Tuned LGB reconstruction. Red Channel: NIR; Green Channel: Red; Blue Channel: Green

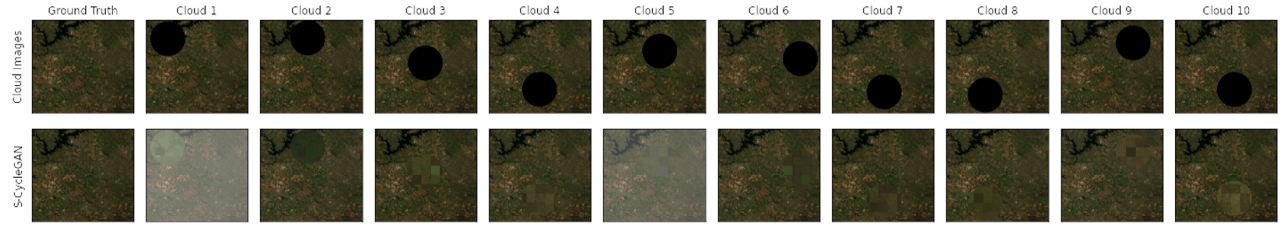


Figure 11: Cloudy Sentinel-2 RGB images reconstructed from Sentinel-1 SAR Data. Panels from top to bottom: Cloudy images, S-CycleGAN [5]. Original images were too dark to easily view, so brightness, exposure, and contrast were edited in Microsoft Photos for each model image (consistently across all models) for visualization. The predictions are inconsistently colored due to the normalization technique used for model building. This model requires tuning.