Internship repport 3A - IRIS

# My work as assistant researcher in the socio-ecological labs of UVic

---

Realise by:
Côme Beauquier

Intership supervisor:
Noémie Boulanger-Lapointe

Referent teacher:
Audrey Minghelli

academic year 2024/2025

# Engagement de non plagiat.

Je soussigné, Côme Beauquier

N° carte d'étudiant : 22106190

Déclare avoir pris connaissance de la charte des examens et notamment du paragraphe spécifique au plagiat.

Je suis pleinement conscient(e) que la copie intégrale sans citation ni référence de documents ou d'une partie de document publiés sous quelques formes que ce soit (ouvrages, publications, rapports d'étudiants, internet, etc….) est un plagiat et constitue une violation des droits d'auteur ainsi qu'une fraude caractérisée.

En conséquence, je m'engage à citer toutes les sources que j'ai utilisées pour produire et écrire ce document.

Fait le 17/07/23

Signature(s)

Ce document doit être inséré en première page de tous les rapports, dossiers et/ou mémoires.

*Document du chapitre 10 annexe 5, issu de la Charte des examens adoptée en Conseil d'Administration le 11 juillet 2013 après avis du CEVU du 27 juin 2013 - Délibération N°2013-73 – Modifié suite au CFVU du 12/03/2015.*

# Contents

# 1 Acknowledgment

First and foremost, I would like to express my special thanks to **Noémie Boulanger-Lapointe**, my supervisor, for trusting me to work on the Garibaldi Park project. She provided invaluable guidance throughout my research process, offering insightful advice and helping me ask the right questions. I am also deeply grateful to **Colleen Dawson**, the Garibaldi project manager, for her assistance in organizing the fieldwork.

Furthermore, I would like to extend my gratitude to the educational and administrative team at Seatech for making this internship possible, with a particular thanks to **Audrey Minghelli** for ensuring everything went smoothly.

Last but not least, I want to express my heartfelt thanks to my family for their unwavering support in my academic and career choices. A special mention to my parents, **Anne Dubois** and **Eric Beauquier**, for her constant encouragement and assistance throughout my studies.

## 2 Resume

For my end-of-studies project, I completed an internship at the University of Victoria in British Columbia, Canada. This report summarizes the last five months I spent in the socio-ecological landscape laboratory. I will try to highlight what my initial goals were, what I accomplished during this time, and what conclusions I drew from this internship.

Before arriving in Canada, I had different expectations:

- Learn more about Alpine and biological research. As a mountain enthusiant, I really wanted to orient myself in this domain.
- Improve my skills in data analysis, especially in image processing.
- Become more comfortable with my English.

During this internship, I actually had a lot of freedom in what I needed to do. The subject of my research was not clearly defined when I arrived. So, one of the first things I did was gather information about the research conducted in this park and try to figure out how I could contribute to the project.

Finally, I explored different subjects and methods for processing satellite images. My main goal was to study the forest in Garibaldi Park. Here are the main tasks I completed:

- **Documentation:** One of the first things I did was document and review the literature on how to create a classification, from data collection to classification evaluation. This was a good step to see my work more clearly.
- **Classification:** I tried to train a supervised classification model to identify the different types of environments in the park.
- **Canopy Cover Map:** I created a canopy cover map of the park and tried to link it with fieldwork data.
- **Fieldwork:** I spent two weeks in the park collecting data.
- **Solar Panel:** I sized and built a solar panel to recharge drone batteries.
- **Georeferencing of Drone Images:** I post-processed drone images taken during field work.

This internship was a bit frustrating for me because five months is a very short time to deeply delve into a research topic, and I am not completely convinced by my results. On the other hand, I learned a lot, especially about exploring topics and not being afraid to go back when I was not on the right path. Being lost in your research is normal and part of the process. Next time I work in research, I think I will take more time to establish a plan by studying similar work instead of immediately trying to obtain results.

# 3 Introduction

The end-of-studies internship is an important moment for transitioning into working life. It allows us to make a smooth transition into the workforce and gain experience in our chosen field. Throughout my studies, I asked myself what I wanted to do after graduation. What I always knew is that I wanted a job with a significant impact on society that is also technically stimulating. Additionally, as a mountain lover, I really wanted to learn more about nature protection and biological studies, as these are subjects that make a lot of sense to me. I also wanted to work in research because I appreciate the freedom it offers, you can explore many different things and are somewhat your own manager.

When I found this laboratory conducting Alpine studies in Garibaldi Provincial Park, it seemed like an obvious choice for me. They work on various topics related to Garibaldi Park and other areas, such as plant surveys.

Research in the Socio-Ecological Landscapes Lab focuses on the impact of climate and anthropogenic changes on plant communities, herbivores, and humans in Arctic and Alpine ecosystems. There are currently two main projects running in this lab: one studying the effects of climate change on berries in the northwest Canadian Territories, and the one I am working on, which examines the impact of climate change and recreational activities on mountain systems.

All the projects in the lab are under the guidance of Dr. Noémie Boulanger-Lapointe, a professor at the University of Victoria in the geography department. But the Garibaldi project is a large research initiative conducted in collaboration with other labs at the University of British Columbia and Simon Fraser University, as well as with British Columbia Parks and the Sḵwx̱wu7mesh Nation.

This project has three main goals:

- Document the Sḵwx̱wu7mesh Nation's past and current relationship with the alpine.
- Describe the current and historical composition of biological communities.
- Evaluate the impact of global changes on alpine ecosystems.

I will contribute to the second goal by trying to collect information on forest cover and the tree line.

# 4 Presentation of the park

Garibaldi Provincial Park is a volcanic mountain massif located north of Vancouver. It is a huge park with an area of over 1,940 square kilometers. It was established in 1920 and classified as a provincial park of British Columbia in 1927. The park is situated within the traditional territory of the Swx̱wú7mesh Nation, the First Nation government of the Squamish people, who have lived there for more than a thousand years. Despite this, the park still bears the name of an Italian man.

This park is very famous as an outdoor recreation destination. In fact, due to its proximity to cities and easy access, it is one of the most popular places for hiking, backcountry camping, mountain biking, and other mountain activities. This makes it an excellent field site for studying the impact of human activities on nature.

However, Garibaldi Park is also a significant ecological reserve. While human activities are currently concentrated in the eastern part of the park, the rest remains untouched wilderness without any human recreation. As a result, the park also boasts a rich diversity of fauna and flora, including forests, plants, birds, and mammals such as mountain goats and bears.

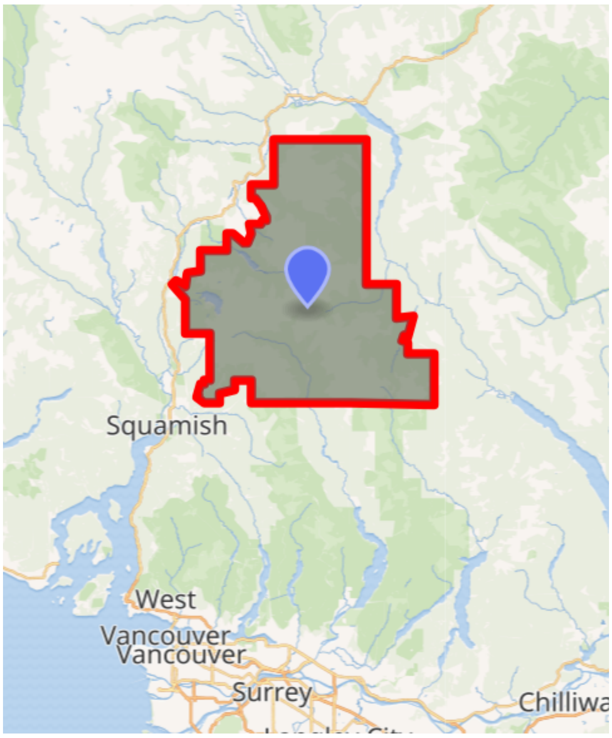



Figure 1: Boundary of the Garibaldi Provincial park

# 5 Image satellite and classification documentation

To begin this report, we will go through the documentation I prepared on satellite imagery and classification. I reviewed many scientific articles to learn about the different methods used and gain an overall understanding of the field.

## 5.1 Image satellite data

Firstly, we will explore the satellite data we will use and the reasons behind our choices. Three satellites provide free data suitable for our study: Sentinel-2, Landsat 9, and PlanetScope. The table below summarizes the properties of each:

| Satellite | Band | Name | Spectral Range (nm) | Spatial Resolution (m) | Revisit Time |
|---|---|---|---|---|---|
| Sentinel-2 | B1 | Coastal aerosol | 433–453 | 60 | 5 days |
| | B2 | Blue | 458–523 | 10 | |
| | B3 | Green | 543–578 | 10 | |
| | B4 | Red | 650–680 | 10 | |
| | B5 | Red Edge 1 | 697–713 | 20 | |
| | B6 | Red Edge 2 | 732–748 | 20 | |
| | B7 | Red Edge 3 | 773–793 | 20 | |
| | B8 | NIR | 785–900 | 10 | |
| | B8A | Narrow NIR | 855–875 | 20 | |
| | B9 | Water vapor | 935–955 | 60 | |
| | B10 | SWIR – Cirrus | 1360–1390 | 60 | |
| | B11 | SWIR 1 | 1565–1655 | 20 | |
| | B12 | SWIR 2 | 2100–2280 | 20 | |
| Landsat 9 | B1 | Coastal/Aerosol | 433–453 | 30 | 8 days |
| | B2 | Blue | 450–515 | 30 | |
| | B3 | Green | 525–600 | 30 | |
| | B4 | Red | 630–680 | 30 | |
| | B5 | NIR | 845–885 | 30 | |
| | B6 | SWIR 1 | 1560–1660 | 30 | |
| | B7 | SWIR 2 | 2100–2300 | 30 | |
| | B8 | Panchromatic | 500–680 | 15 | |
| | B9 | Cirrus | 1360–1390 | 30 | |
| | B10 | Thermal IR 1 | 10300–11300 | 100 | |
| | B11 | Thermal IR 2 | 11500–12500 | 100 | |
| PlanetScope | B1 | Coastal Blue | 431–452 | 3 | 1 day |
| | B2 | Blue | 465–515 | 3 | |
| | B3 | Green I | 513–549 | 3 | |
| | B4 | Green | 547–583 | 3 | |
| | B5 | Yellow | 600–620 | 3 | |
| | B6 | Red | 650–680 | 3 | |
| | B7 | Red Edge | 697–713 | 3 | |
| | B8 | NIR | 845–885 | 3 | |

Table 1: Comparison of spectral, spatial, and temporal characteristics of Sentinel-2, Landsat 9, and PlanetScope (SuperDove) satellite sensors.

When choosing satellite data, there are a few key factors to consider:

- **Spatial Resolution:** The higher the resolution, the more precise our classification results will be. It also allows us to detect smaller features in the landscape. For example, in the study by Flores et al. [2024], PlanetScope data was used to detect tiny proglacial headwaters downstream—something only possible with its 3-meter resolution.

- **Spectral Bands:** The available spectral bands are critical for analyzing alpine vegetation. The following bands are particularly useful:
  - **Coastal/Blue (≈ 450 nm):** Highlights areas with high reflectance, such as snow, glaciers, and water.
  - **Green (≈ 550 nm):** Reflects chlorophyll, used to detect vegetation through photosynthesis.
  - **Red (≈ 655 nm):** Absorbed by chlorophyll, this band is essential for vegetation indices like NDVI.
  - **Red Edge (≈ 750 nm):** Useful for differentiating vegetation types and identifying plant stress.
  - **NIR (≈ 850 nm):** Strongly reflected by healthy vegetation, used in NDVI and health assessment.
  - **SWIR (≈ 1600/2200 nm):** Absorbed by water and wet snow, helpful for detecting moisture content.

  Sentinel-2 stands out by offering all the relevant wavelengths and three Red Edge bands, making it especially powerful for identifying diverse vegetation types.
- **Revisit Time:** This is important when studying temporal changes. A shorter revisit time increases the chances of obtaining cloud-free images and monitoring events over time.

In conclusion, Sentinel-2 and PlanetScope appear to be the most suitable options. Sentinel-2 offers a rich spectral range ideal for detailed analysis, while PlanetScope provides superior spatial resolution. As suggested in Li et al. [2020], it may be beneficial to combine the strengths of both datasets—for example, by generating synthetic Red Edge and SWIR bands for PlanetScope using Sentinel-2 data, or by injecting PlanetScope's spatial details into Sentinel-2 imagery. This fusion could significantly enhance the reliability and precision of our dataset.

## 5.2 Additional Data Sources

In addition to satellite imagery, we can incorporate other types of data to improve classification results, such as elevation data, LiDAR, and texture features. Let's explore how each of these can enhance our analysis:

- **LiDAR Data:**
  LiDAR data consist of 3D point clouds representing the Earth's surface with very high resolution. These points are generated from laser reflections off the ground and can include multiple returns (e.g., when the laser penetrates semi-transparent materials). From LiDAR data, we can derive several useful models:
  - DEM (Digital Elevation Model): Represents the bare-earth elevation, excluding vegetation and buildings.
  - DSM (Digital Surface Model): Includes the elevation of all surface features, such as vegetation and buildings.
  - CHM (Canopy Height Model): Represents vegetation height by subtracting the DEM from the DSM. This is particularly useful for identifying forests.

According to Reese et al. [2014], incorporating LiDAR data significantly improves classification performance, especially in distinguishing similar classes such as low-density forest and grasslands. Unfortunately, LiDAR data is not available for Garibaldi Park, so we will explore alternative approaches.

- **Elevation Data:**
  In the absence of LiDAR, we can use a DEM (Digital Elevation Model) to incorporate elevation into our analysis. In Wu et al. [2005], the authors explored different methods for integrating DEM data into satellite imagery. They found that the most effective approach was to simply add the DEM as an additional raster band during preprocessing. This significantly improved classification accuracy, particularly in alpine environments where elevation and slope have a strong influence on land cover types.

Another way to enhance classification is by extracting features directly from the imagery, such as texture or vegetation indices. In Mohammadpour et al. [2022], the authors improved classification accuracy by computing several vegetation indices: NDVI (Normalized Difference Vegetation Index), GNDVI (Green NDVI), EVI (Enhanced Vegetation Index), and SAVI (Soil-Adjusted Vegetation Index). They also extracted image textures using Principal Component Analysis (PCA). Texture features help distinguish between land cover types—for example, forests tend to have more texture than glaciers—and they help define more precise class boundaries. Both vegetation indices and texture analysis significantly improved the performance of classification models, especially for vegetation-related classes.

## 5.3 Classification

There are different methods we can use for land cover classification. Let's explore some of the most common ones, along with their strengths and weaknesses.

- **Supervised Classification:**
  This is probably the most commonly used method. It relies on a set of already labeled data to train a model, such as Random Forest or Support Vector Machines (SVM). If the labeled data is accurate and representative, this approach can yield highly precise results and capture complex interactions between features. However, the quality of the results depends heavily on the quality of the training data.

- **Unsupervised Classification:**
  This method does not require labeled data. Instead, it automatically groups pixels into clusters based on similarity, with the number of clusters defined by the user. While generally less precise than supervised methods, unsupervised classification can be useful for identifying hidden patterns or unknown classes. It typically requires post-processing to interpret and label the clusters. This makes it useful for preliminary studies or exploratory analysis.

Supervised classification appears to be the most suitable approach for our case. However, this raises a key question: how can we create a good labeled dataset?

## 5.4 Training datas

The review by Tra provides an excellent overview of techniques for building training datasets and training models. Let's look at the main options:

- **Manual Collection:**
  This involves identifying representative points or zones for each target class using high-resolution imagery (e.g., Google Earth) or GPS points collected in the field. The downside of this method is the time required—especially for ground surveys—and the potential for subjective interpretation errors.

- **Automatic Collection:**
  This approach uses existing land cover datasets as sources of training data. The main advantage is that it enables the efficient creation of large training datasets. However, the quality of the results depends on how well the dataset matches the specific characteristics of the study area. Many existing datasets are highly generalized (e.g., they do not distinguish between different types of forests). By relying on these datasets, you become dependent on their classification schemes, so it is essential to verify that the classes align well with the local environmental conditions.

- **Hybrid Collection:**
  As demonstrated in Cla, a hybrid approach can be very effective. It consists of starting from an existing dataset, selecting and adapting relevant classes, and supplementing or refining them manually. While this method also carries the risk of interpretation errors and can be time-consuming (depending on the base data quality), it often provides a good balance between efficiency and precision.

There is no one-size-fits-all solution for creating training data. If a high-quality dataset specific to your region already exists, automatic collection might be enough. However, in most cases, manual or hybrid methods are necessary. Regardless of the method, the more precise and thoughtful the labeling process, the better the final classification.

In the case of Garibaldi Park, the best approach seems to be a hybrid data collection. Several global datasets, such as ESA WorldCover, already offer valuable baseline information. The idea is to download these datasets, verify their consistency with the high-resolution drone imagery, and enhance them—for example, by distinguishing between ice and snow or between compact rock and scree slopes.

## 5.5 Pre-processing of Image Data

Before training the model, the image data must undergo pre-processing to ensure optimal quality. There are three main types of corrections:

- **Radiometric Correction:** This corrects variations caused by the sensor, such as camera noise, sun glare, or electrical interference.
- **Atmospheric Correction:** This addresses distortions caused by the atmosphere, such as those caused by aerosols, water vapor and other atmospheric particles that affect light transmission.
- **Geometric Correction:** This ensures that the image is accurately georeferenced by correcting distortions caused by satellite inclination, terrain elevation, and pixel misalignments.

Most of these corrections are already applied to satellite images before they are made available for download. This is the case for both PlanetScope and Sentinel-2 images. However, it's important to be aware of these corrections, as further adjustments—particularly atmospheric corrections—can be added to improve precision when needed. Both datasets also provide cloud and shadow masks, which can be used to exclude those areas during processing.

### 5.6 Training Classification Models

Now that we have prepared all the data, it's time to train our classification model. Many supervised classification algorithms exist, but in the literature, three models appear most frequently: Decision Trees (DT), Support Vector Machines (SVM), and Convolutional Neural Networks (CNN). Let's explore how these models work:

- **Decision Trees (DT):** These are classification and regression methods used in machine learning. The model builds a tree structure where each node represents a decision—for example, "Is the pixel value in Band 1 greater than threshold $x$?" The branches lead to further decisions until reaching a leaf node, which assigns a class to the pixel. More advanced algorithms like Random Forest build multiple decision trees and combine their results to improve accuracy. The main advantages of DTs are their simplicity and strong performance.
- **Support Vector Machines (SVM):** SVMs are also used for classification and regression. They work by finding the optimal hyperplane that separates classes in a high-dimensional feature space (where features include spectral bands, elevation, indices, etc.). SVMs generally produce very good results, but they are more complex and computationally intensive than Random Forests.
- **Convolutional Neural Networks (CNN):** CNNs use multiple layers of interconnected neurons to learn how to classify pixels. They are capable of achieving highly accurate results, especially with spatial patterns, but require large training datasets and significant computational power for training.

All three models are effective for handling complex, high-dimensional classification tasks. They can even be combined to leverage their individual strengths. For example, in Zafari et al. [2019], the authors combined Random Forest and SVM models, achieving better results than using standard Random Forest alone in high-dimensional settings.

Since implementing a CNN is more complex and resource intensive, the best approach for our case might be to experiment with both Random Forest and SVM, and possibly a combination of the two.

### 5.7 How to Evaluate Our Classification

To evaluate the performance of a classification model, the most common approach is to split the dataset into two parts: one for training and one for testing the model. Usually, a larger portion is used for training (e.g., 80% for training and 20% for testing). Using the test data, we can identify correct and incorrect predictions and thus measure the model's performance. These data are chose randomly.

To evaluate performance across all classes, we can compute a global confusion matrix that summarizes true and false positives for each class. For example, for three classes (A, B, and C), the matrix has the following form:

| | **Predicted A** | **Predicted B** | **Predicted C** |
|---|---|---|---|
| **Actual A** | TP_AA | FP_BA | FP_CA |
| **Actual B** | FP_AB | TP_BB | FP_CB |
| **Actual C** | FP_AC | FP_BC | TP_CC |

Table 2: Confusion Matrix

Where:

- TP_AA: True Positives — number of pixels correctly classified as class A
- FP_AB: False Positives — number of pixels classified as A but actually belong to class B

We can then compute the total false positives for a class. For example, for class A:

$$\text{FP}_A = \text{FP}_{AB} + \text{FP}_{AC}$$

From the confusion matrix, we can also derive:

- False Negatives for class A: $\text{FN}_A = \text{FP}_{BA} + \text{FP}_{CA}$
- True Negatives for class A: $\text{TN}_A = \text{TP}_{BB} + \text{FP}_{CB} + \text{FP}_{BC} + \text{TP}_{CC}$

This matrix helps us identify which classes are often confused with others. Additionally, we can compute several standard metrics to assess class-wise performance. As shown in Flores et al. [2024], these include:

- **Precision** = $\frac{TP}{TP+FP}$ — the proportion of correct positive predictions among all positive predictions
- **Recall** = $\frac{TP}{TP+FN}$ — the proportion of actual positives that were correctly predicted
- **Specificity** = $\frac{TN}{TN+FP}$ — the proportion of actual negatives that were correctly predicted
- **F1-score** = $\frac{2\cdot(\text{Precision}\cdot\text{Recall})}{\text{Precision}+\text{Recall}}$ — the harmonic mean of Precision and Recall

This allows us to see the performance of our model for each class. However, we can also use a metric to get an idea of the overall performance of our model. In the article Zafari et al. [2019], the following tool is used:

- Overall Accuracy (OA) = $\frac{\sum TP}{\sum \text{Predictions}}$: Proportion of true positives (TP) relative to the total number of predictions. This is an easy way to evaluate the model.

This metric is useful for assessing the overall performance of our model.

# 6 Making the classification

After this extensive documentation, let's see how I used this information in my work. My first goal was to create a simple classification of the park area.

To perform this classification, I initially decided to use PlanetScope data (cf 1) due to its high precision of 3 meters. However, this also means it is computationally demanding. For these calculations, I used supercomputers hosted by the Digital Research Alliance of Canada. Let's see how I did it.

## 6.1 Computing big data

The Digital Research Alliance of Canada provides Canadian researchers with access to storage and computational power. By running your code on Compute Canada, you can select the number of CPUs and the amount of RAM allocated to each CPU. I created a tutorial on how to use these supercomputers, which you can find in the annex for more details (1).

For my work, I used a parallelization method to process my images across multiple CPUs simultaneously. Here's how I did it:

To implement parallelization, the idea is to divide the main raster (pixel image) into smaller raster blocks based on the number of CPUs being used. Each CPU processes its assigned raster block in parallel (simultaneously), and then all the processed raster blocks are merged back together to reconstruct the entire raster.

To distribute the raster blocks across the CPUs, I used the "doParallel" package in R. For more information about this, you can check out the commented code on GitHub.

## 6.2 Unsupervised Classification

I first attempted an unsupervised classification of the park to gain insight into the different types of land cover present. For this, I used the K-means method. The idea behind this algorithm is to classify the pixels of my raster by grouping them into clusters. The algorithm identifies these clusters in a way that minimizes the variance between each pixel and the centroid of its assigned cluster. Here is a simplified example to illustrate this method:

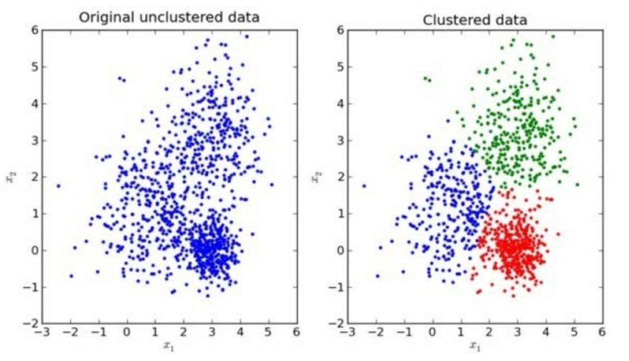



Figure 2: K-means algorithm illustration with 3 clusters

In our case, the algorithm considers all the bands of my raster.

Here is an illustration of a small portion of the park classified into 6 clusters:

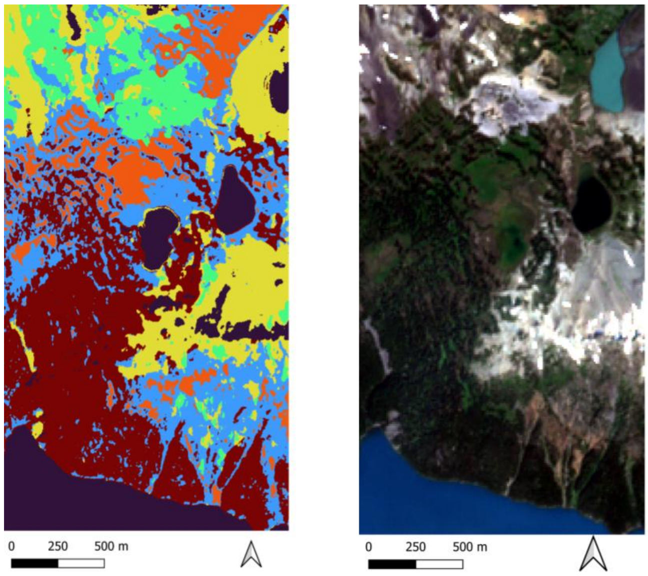



Figure 3: K-means unsupervised classification map

After examining maps of the park with different numbers of clusters, I identified 7 main types of land cover:

- Alpine Vegetation
- Bare Land
- Forest
- Rock
- Snow
- Glacier
- Water

Based on these results, I decided to proceed with supervised classification to achieve more precise outcomes.

## 6.3 Training supervised classification model

For supervised classification, I trained a decision tree model (cf 5.6). For this, I used the Ranger package, which is designed to efficiently build Random Forest models.

To build a model, the 'ranger' function requires a training dataset and the number of decision trees to construct. To train this model, I needed training data. Here's how I constructed my training datasets:

- **Manual collection:** First, I created a supervised classification by manually building the dataset. I drew polygons on real-color satellite images of the park, creating 2-3 polygons per field category (as listed in 6.2). All pixels within these polygons formed my dataset.

- **Automatic collection:** For this, I used the ESA WorldCover dataset, a global classification from 2021. The classes in this dataset include: Tree cover, Shrubland, Grassland, Cropland, Built-up, Bare/sparse vegetation, Snow and ice, Permanent water bodies, Herbaceous wetland, Mangroves, and Moss and lichen. To build my training samples, I randomly selected $n$ samples from each class within the park boundaries. If a class had fewer than $n$ pixels in the area, I used all available pixels from that class.

Once I had my dataset, I split it into training and validation datasets, typically using 80% of the data for training and 20% for validation. I then evaluated the performance of my model by calculating the confusion matrix, overall accuracy, and for each class: Precision, Recall, Specificity, and F1-score (cf 5.7).

### Results and discussion

Let's first examine the results for my manual collection classification. Here is the confusion matrix and other performance metrics:

| Predicted | bareLand | alpineVegetation | forest | ice | rock | snow | water |
|---|---|---|---|---|---|---|---|
| bareLand | 10865 | 0 | 2 | 0 | 41 | 0 | 0 |
| alpineVegetation | 0 | 3318 | 29 | 0 | 0 | 0 | 0 |
| forest | 15 | 83 | 106720 | 0 | 0 | 0 | 0 |
| ice | 0 | 0 | 0 | 25442 | 6 | 9 | 0 |
| rock | 46 | 0 | 1 | 41 | 30787 | 0 | 0 |
| snow | 0 | 0 | 0 | 20 | 1 | 67186 | 0 |
| water | 0 | 0 | 0 | 0 | 0 | 0 | 34562 |
| Precision | 0.994 | 0.975 | 0.999 | 0.997 | 0.998 | 0.999 | 1 |
| Recall | 0.996 | 0.991 | 0.999 | 0.999 | 0.997 | 0.999 | 1 |
| F1-score | 0.995 | 0.983 | 0.999 | 0.998 | 0.997 | 0.999 | 1 |

Table 3: Confusion matrix of manual collection training set

The overall accuracy is $OA = 0.998$. While these results appear excellent, we must consider potential biases. The training and test sets were derived from the same polygons in the same environment.

Here is the classification map:

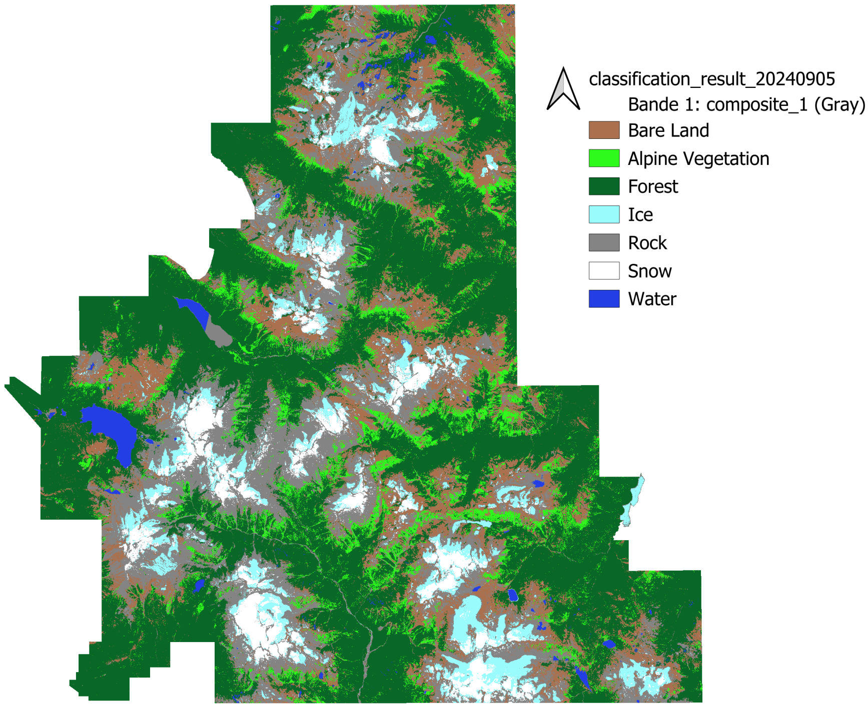
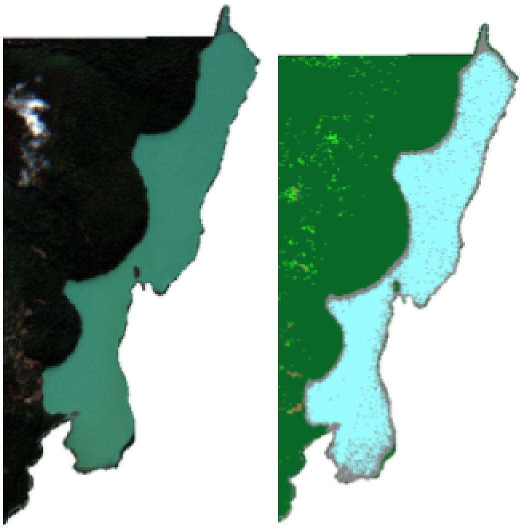


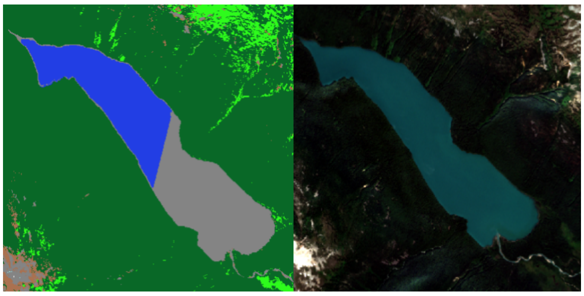

Figure 4: Classification with manual collection

Overall, the classification appears good, with each class generally located where expected. However, upon closer inspection, several issues become apparent. There is notable confusion between: - Ice and Water/Rock - Rock and Bare Land - Forest and Alpine Vegetation - Water and Rock

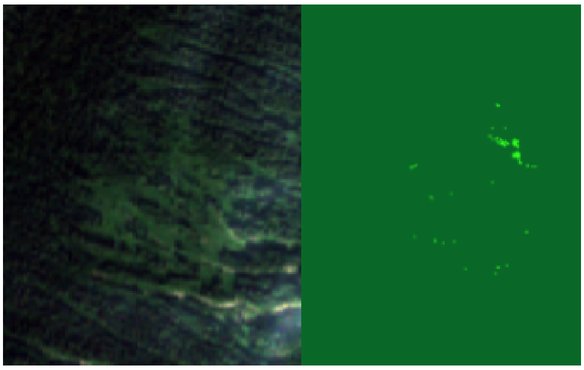
Here are some examples of these confusions:



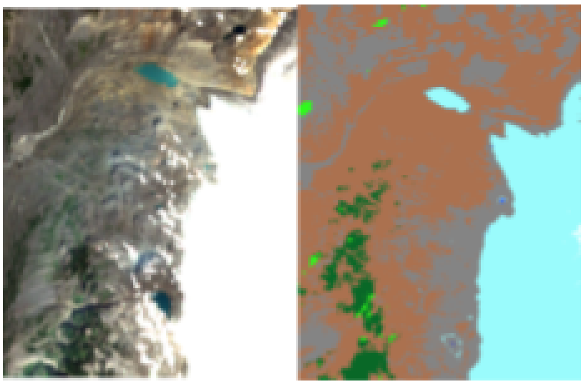
(a) Between water and ice



(b) Between water and rock



(c) Between alpine vegetation and forest



(d) Between bare land and rock

Figure 5: Confusion examples in manual classification

Upon closer examination, it's clear that the classification is not as accurate as the overall accuracy suggests. The confusion in different areas of the park is evident. In image (b), the confusion between water and rock shows a distinct line, which is due to the merging of satellite images. Even when using the same satellite, there can be differences between images.

These confusions are not due to the model itself but rather to how the training dataset was constructed. The selected areas for each class were not fully representative of the diversity across the entire park. To address this, I will try using random reference points throughout the park to see if this improves the results.

To avoid these biases, I attempted to perform the classification using an automatic collection with the ESA WorldCover dataset, as explained in 6.3. Since the points are selected randomly, this approach should reduce bias, but the biases inherent in the ESA classification remain. I constructed my dataset by randomly selecting 100,000 pixels (or all pixels if the class had fewer) from each class. After training my model, here are the metrics I obtained:

Table 4: Confusion matrix

| Reference | T.Cov | Mang | Grass | Bare | S/Ice | Water | Shrub | H.Wet | B.Up |
|---|---|---|---|---|---|---|---|---|---|
| Tree Cover | 27432 | 336 | 554 | 263 | 160 | 186 | 35 | 0 | 0 |
| Mangroves | 131 | 23778 | 212 | 140 | 15 | 117 | 1 | 0 | 0 |
| Grassland | 84 | 124 | 24713 | 67 | 11 | 23 | 12 | 0 | 0 |
| Bare | 114 | 159 | 139 | 22585 | 182 | 132 | 24 | 0 | 0 |
| Snow/Ice | 587 | 690 | 690 | 839 | 23637 | 290 | 81 | 0 | 0 |
| Water | 739 | 902 | 838 | 665 | 317 | 41339 | 59 | 0 | 0 |
| Shrubland | 0 | 0 | 0 | 0 | 0 | 0 | 1759 | 0 | 0 |
| Herbaceous wetland | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 46 | 0 |
| Built-up | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 28 |

This gives us a global accuracy of **94.34%**.

The results are less accurate than those from the manual collection, which is logical since the points were selected randomly. Nevertheless, these results are still very good.

However, issues arise with the classification itself. I was unable to produce a classification that made sense with this model. Here is an example of what I obtained:

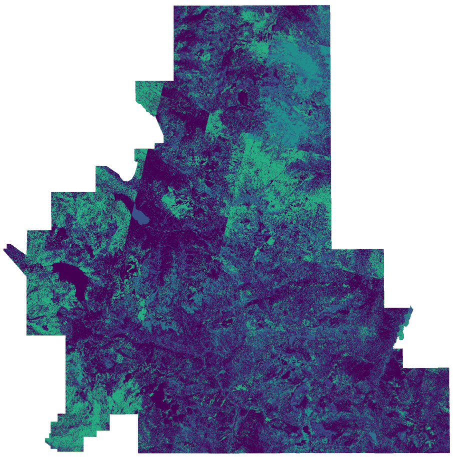


Figure 6: Example of automatic collection classification using ESA WorldCover data

While lakes, glaciers, and valleys are sometimes recognizable, most of the map appears random. Additionally, the merging of different satellite images is visible, and our classification is highly sensitive to these artifacts. I have not been able to identify what is going wrong with this classification. My goal was to use the ESA data to correct the biases of the manual collection classification.

After further documentation, I finally decided to focus more on spectral indices to try to obtain better results.

# 7 Canopy density map

Finally, I decided to focus more on vegetation indices and other classic methods. First, I used one of the additional datasets to separate vegetation from other land types. For this, I employed the NDVI (Normalized Difference Vegetation Index).

## 7.1 Discrimination of vegetation

To calculate the NDVI, we use two bands: the NIR (Near-Infrared) and Red bands. Vegetation absorbs red light during photosynthesis and reflects or emits strongly in the NIR band. To detect vegetation, we calculate the difference between the reflectance of NIR (which should be high) and Red (which should be low), and then normalize it. The formula is as follows:

$$NDVI = \frac{NIR-Red}{NIR+Red}$$

Due to the normalization, NDVI values range between -1 and 1. According to the USGS (United States Geological Survey) website, here are the principal ranges for different land types:

- $NDVI \lessapprox 0.1$: Areas of barren rock, sand, or snow typically exhibit very low NDVI values.
- $0.2 \lessapprox NDVI \lessapprox 0.5$: Sparse vegetation, such as shrubs, grasslands, or senescing crops, usually results in moderate NDVI values.
- $NDVI \gtrapprox 0.6$: Corresponds to dense vegetation, such as that found in temperate and tropical forests or crops at their peak growth stage.

To discriminate vegetation, I used Sentinel-2 image data. Since vegetation typically covers broad areas rather than isolated points, high spatial resolution is not essential for this analysis. Given this consideration, Sentinel-2 data is more suitable than PlanetScope data due to the reduced computational complexity of the calculations.

According to Table 1, the NDVI calculation using Sentinel-2 bands is as follows:

$$NDVI = \frac{B8-B4}{B8+B4}$$

We then obtained a raster containing all the NDVI values. To analyze the distribution of these values and determine the threshold for discriminating vegetation, I generated a histogram:

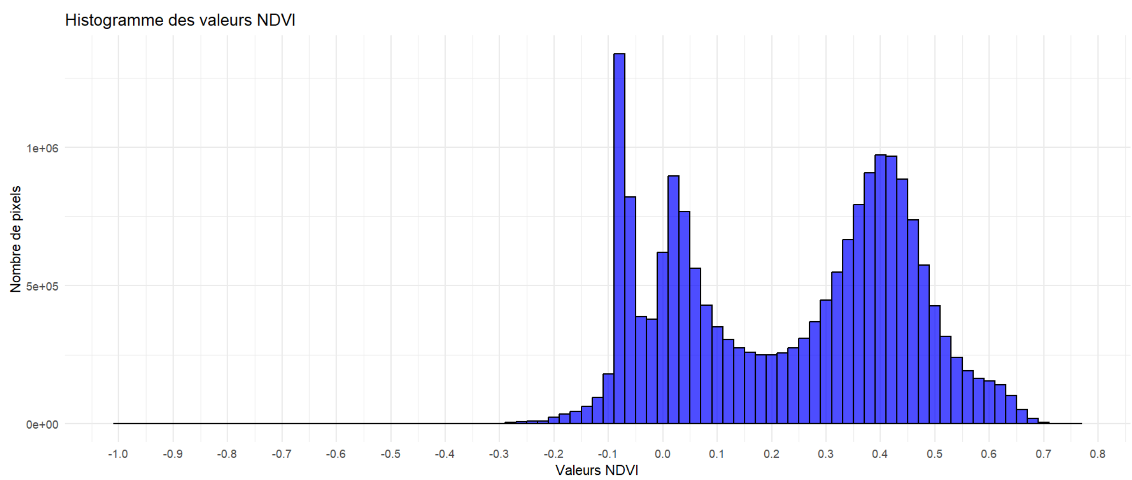
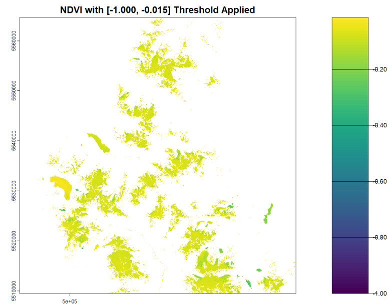


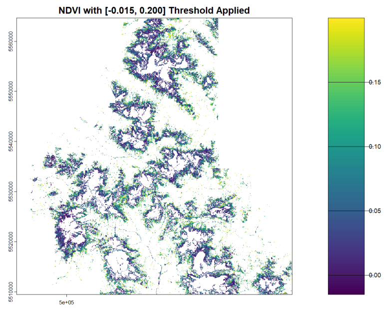

Figure 7: NDVI value distribution histogram

The histogram clearly shows three peaks: one around -0.1, which likely corresponds to glaciers, snow, and lakes; one around 0, which should represent rocky areas; and one around 0.4, which is likely vegetation. Let's apply thresholds to the NDVI raster to confirm the classification of these peaks.

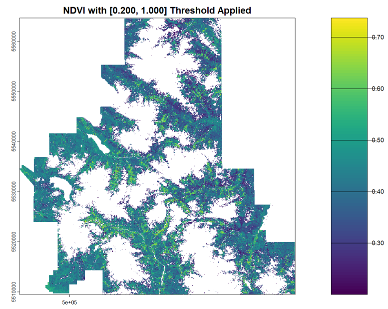
(a) Recomposed image




(b) NDVI with low threshold




(c) NDVI with middle threshold




(d) NDVI with high threshold

Figure 8: NDVI thresholding results

By superimposing and comparing the NDVI raster with the recomposed image, we adjusted the thresholds to achieve the best results:

- **Glacier/Rock-Dirt threshold:** We decided to raise the threshold slightly because some parts of the lakes were being classified as rock. However, doing so caused rock glaciers to start being classified as rock. This issue could potentially be resolved using other indices such as NDSI (Normalized Difference Snow Index) or NDWI (Normalized Difference Water Index), but for now, we will focus on vegetation.

- **Rock-Dirt/Vegetation threshold:** The threshold of 0.2 effectively separated rock-dirt areas from vegetation.

One remaining issue is the presence of rare clouds and mountain shadows. Preprocessing could be useful to address these artifacts.

## 7.2 Conversion of Sentinel-2 data into reflectance values

The Sentinel-2 L2A data are already normalized and multiplied by a quantification factor of 10,000. To use these data, I divided them by 10,000 and then multiplied by 255 to scale the values between 0 and 255. This range is standard for image rasters and allows me to easily apply formulas found in the literature.

However, upon examining the data, it is evident that the values range between 0 and 21,000 for each band, rather than between 0 and 10,000. This is due to high-reflectance surfaces such as snow or clouds, which overexpose the sensors. These overexposed values can be truncated, as vegetation does not exhibit such high reflectance.

Figure 9: Overexposed values in Sentinel-2 data for band 1

The overexposed values are in the minority. When comparing these rasters with the recomposed image, it is clear that these values primarily originate from clouds, with some coming from glaciers.

### 7.3 Forest Canopy Density

We will now use different indices to create a Canopy Density map.

The tree canopy is the layer of branches and leaves that cover the ground when viewed from above. Let's examine how we calculated it.

According to the studies by Sahana et al. [2015] and FOR [2008], three different indices were used to construct the Forest Canopy Density map (all bands are normalized between 0 and 255):

- **AVI** (Advanced Vegetation Index): This is an enhancement of the NDVI. It also uses the NIR and Red bands for calculation, but instead of a relative difference between the bands, it applies a non-linear transformation:

$$AVI = ((NIR + 1)(256 - Red)(NIR - Red))^{1/3}$$
$$AVI = 0 \text{ if } Red \geq NIR$$

  This method allows us to highlight variations in vegetation density.

- **BSI** (Bare Soil Index): This index helps distinguish between bare ground, sparse canopy, and dense canopy. It is calculated as the difference between the sum of two reflective bands and two absorption bands:

$$BSI = \frac{(SWIR1+Red)-(NIR+Blue)}{(SWIR1+Red)+(NIR+Blue)}$$

- **SI** (Shadow Index): This index uses the visible bands of the spectrum to measure the energy that is not reflected back to the sensor. Less reflected energy indicates more shadows, suggesting a denser canopy:

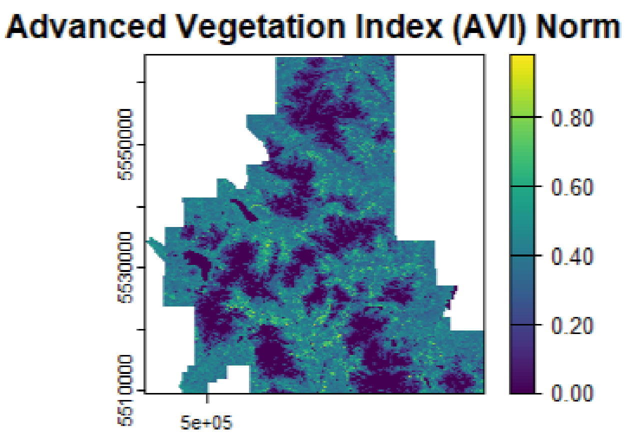
$$SI = ((256 - Blue)(256 - Red)(256 - Green))^{1/3}$$

According to FOR [2008], the relationship between the indices and their values under different forest canopy densities is as follows:

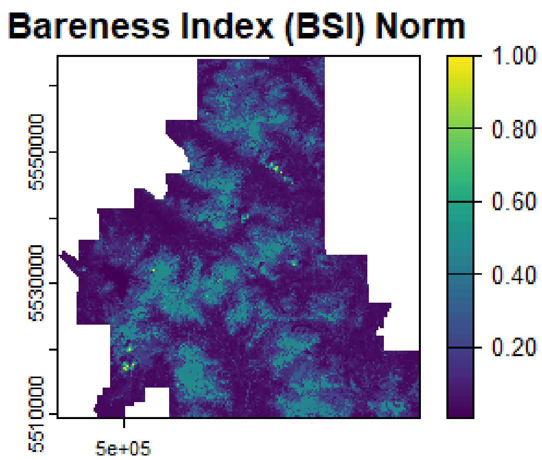
| | Hi-FCD | Mid-FCD | Low-FCD | Grassland | Bare Land |
|---|---|---|---|---|---|
| AVI | High | High | Medium | High | Low |
| BSI | Low | Low | Low | Medium | High |
| SI | High | High | Medium | Low | Low |

Table 5: Combination characteristics between tree indices

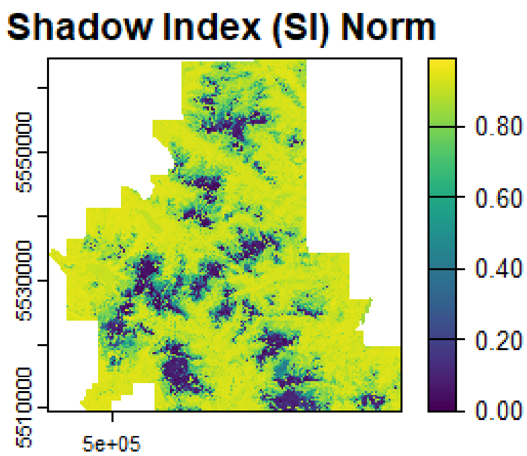
The goal now is to combine all these rasters to create a forest canopy density map. Let's examine the different index maps:




(a) AVI (Advanced Vegetation Index)




(b) BSI (Bare Soil Index)




(c) SI (Shadow Index)

Figure 10: Index maps for forest canopy density

Overall, these maps allow us to draw the following conclusions:

- **AVI:** The AVI values are higher in forested areas than in rocky or glacial areas, which is consistent with expectations.
- **SI:** The Shadow Index (SI) is very high in valleys and very low in mountainous areas (glaciers and rocks). However, the large gap between these values makes the SI less precise for distinguishing between grassland and forest. A potential solution could be to normalize the SI specifically for vegetated areas.
- **BSI:** The Bare Soil Index (BSI) values are lower in forests than in bare mountainous areas, as expected. However, all values are quite low due to the presence of rare clouds, which have very high values. A solution to this issue could be to identify and mask these clouds.

### Calculation of vegetation density

The first step in creating a forest canopy density map is to generate a vegetation density map.

To create this map, we will use two indices: the AVI (Advanced Vegetation Index) and the BSI (Bare Soil Index). These two maps exhibit a high negative correlation. The goal of the vegetation density map is to combine these two indices effectively.

To determine the correct coefficients for combining AVI and BSI, we will perform index integration using Principal Component Analysis (PCA). We are particularly interested in the first principal component (PC1), as it captures most of the variance/covariance of our model when there is a high negative correlation. The equation for PC1 is as follows:

$$PC1 = a \cdot AVI + b \cdot BSI$$

The coefficients $a$ and $b$ are chosen to maximize the variance of PC1. The vegetation density is then equal to PC1. This method allows us to optimally combine the AVI and BSI indices without significant loss of information.

Next, we normalize the Vegetation Density (VD) to express it as a percentage:

$$VD = \frac{\text{value}-\text{min}}{\text{max}-\text{min}} \times 100$$

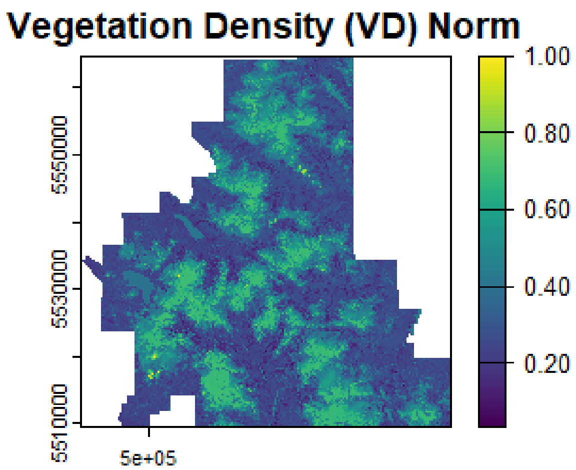



Figure 11: Vegetation Density map

In the vegetation density map, we observe that bare soil areas sometimes exhibit higher values than vegetated areas, which is counterintuitive for a vegetation density metric. I have not been able to identify the exact source of this issue, but it likely stems from the way the AVI and BSI indices were merged using PCA. Another possible cause is the influence of clouds, which have very high BSI values that may dominate and distort other values, particularly during normalization.

### Forest Canopy Density

Despite the issues observed, let's proceed to generate the Forest Canopy Density (FCD) map.

To calculate the FCD, we need both the Vegetation Density (VD) map and the Scaled Shadow Index (SSI) map. The SSI is derived by integrating the Shadow Index (SI), which we discussed earlier, and the Thermal Index (TI). The TI can be calculated using the spectral radiance band from Landsat 5 TM data. Although I had access to this data, time constraints led me to use the normalized SI as a substitute for SSI. According to Sahana et al. [2015], SSI is a linear transformation of SI, and incorporating TI could potentially improve the accuracy of the results. However, using only SI is a reasonable approximation.

The formula for calculating FCD, as provided by Sahana et al. [2015], is as follows:

$$FCD = (VD \times SSI + 1)^{1/2} - 1 \tag{1}$$

Here, both VD and SSI are normalized between 0 and 1. The resulting FCD raster is shown below:

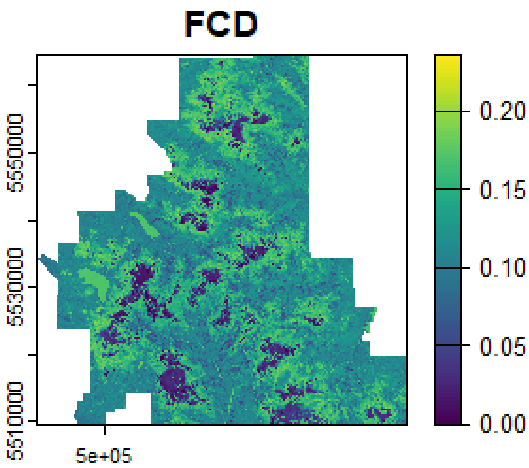



Figure 12: Forest Canopy Density map

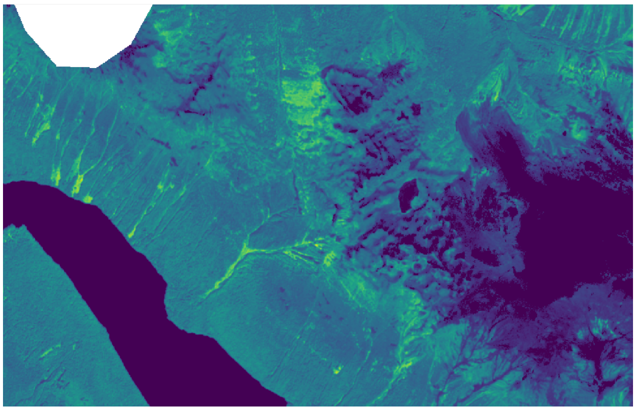
Given that the vegetation density map is not accurate, the forest density map is also unreliable. Let's examine these maps in more detail for a specific area to understand how they work together, what we expect to see, and what we observe in reality.

Let's take this area as an example:

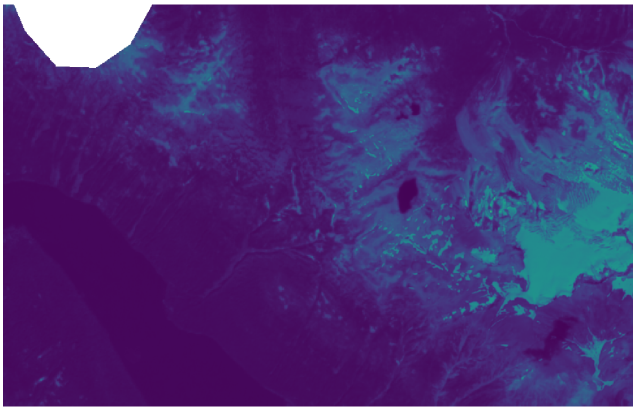
Figure 13: Study area near Whistler ski station

This area is located near Whistler ski station. The lake visible in the image is Cheakamus Lake. I selected this area because it features a variety of landscapes, including lakes, grasslands, forests, glaciers, bare land, and rocks.

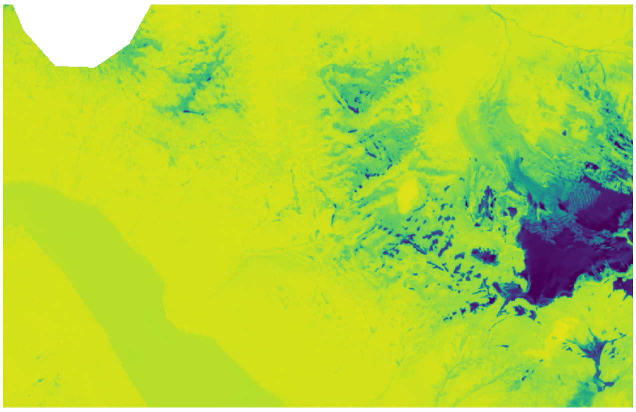
Let's examine and compare the different indices—AVI, BSI, and SI—in this area:



(a) AVI (Advanced Vegetation Index)



(b) BSI (Bare Soil Index)



(c) SI (Shadow Index)

Figure 14: Comparison of AVI, BSI, and SI indices in the study area

- **AVI:** In the AVI map, the index is very low for water and glaciers. It is also low for bare land. The highest values are observed for alpine vegetation and grasslands, while intermediate values correspond to forested areas.

- **BSI:** In the BSI map, all values are quite low, likely due to the influence of clouds, which have very high values. However, glaciers, rocks, and bare land exhibit the highest values. Alpine vegetation and grasslands have lower values, but still higher than those for forests and lakes. It is notable that, except for grasslands and water, BSI values are generally the opposite of AVI values.
- **SI:** It is clear that the values are very low for glaciers and snow. Bare land and rocks also have relatively low values. Water has moderately high values, but significantly lower than those for forests. There is a slight difference between grasslands and forests, though this difference is not substantial.

The goal of the vegetation density map is to integrate AVI and BSI into a single map that encapsulates information about both bare soil and vegetation. This approach should highlight areas with high values for forests and alpine vegetation, while providing a clear distinction between grasslands, moderately dense forests, and highly dense forests. It should also emphasize low values for bare soil areas without vegetation.

Here is the vegetation density (VD) map for this area:

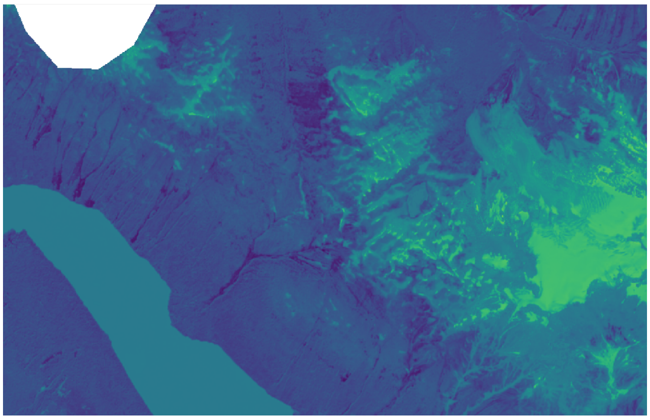


Figure 15: Vegetation Density map

At first glance, it is evident that something is incorrect. Glaciers exhibit very high values, forests have intermediate values, and grasslands have very low values, which is not the desired outcome. Additionally, it makes no sense for water to have high values. I also attempted to combine AVI and BSI using my own factor values.

Here is an example with $a = 1$ and $b = -1$ applied to the same area:

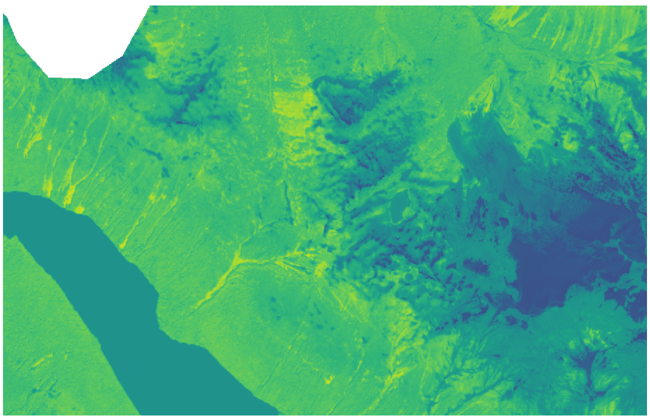
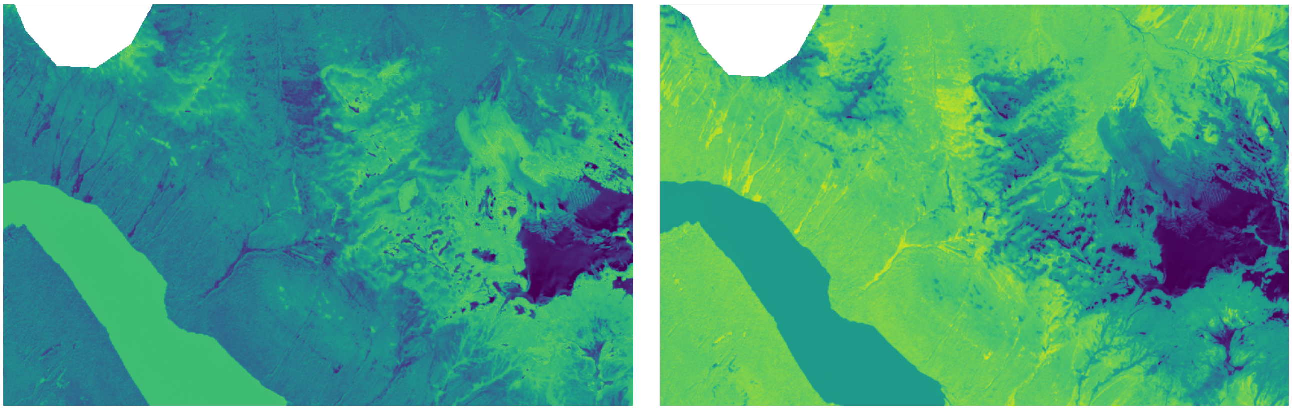


Figure 16: Manually calculated Vegetation Density map

This map appears more logical, with high values for forests and vegetation, and low values for bare land, glaciers, and water. However, there is still an issue: grasslands and areas with short vegetation have higher values than forests, which is not what we want.

Now, let's examine the Forest Canopy Density (FCD):



(a) FCD calculated with PCA-based VD (b) FCD calculated with manually adjusted VD

Figure 17: Forest Canopy Density maps

In the FCD map calculated with the PCA-based VD, vegetation has lower values than water or bare land, which is incorrect.

With the manually calculated VD, the FCD map is more reasonable, as vegetation has higher values than other landscapes. However, there is still an issue: grasslands have higher values than they should. It seems as though the values for vegetation are inverted.

## 7.4 Integration of data

Despite the limitations of my maps, I attempted to associate the canopy cover data, which I collected in the field, with my map.

Here is the FCD map with the points from the fieldwork:

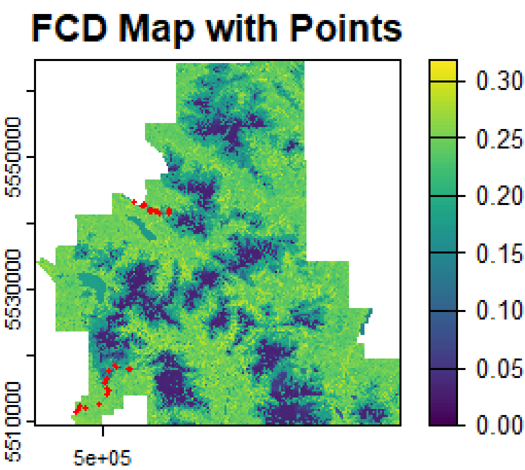



Figure 18: FCD map with points collected during fieldwork

I extracted the FCD values at these points from the map and plotted the FCD values as a function of my densiometer values. Here are the results:

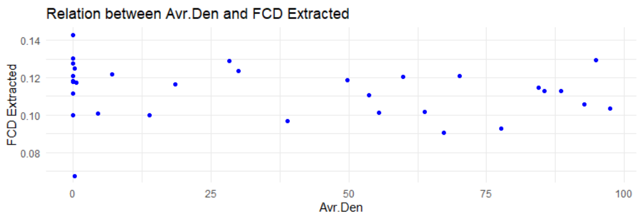



Figure 19: FCD (PCA) values as a function of average densiometer values

No clear trend or relationship is visible between the values, possibly due to the inaccuracies in the FCD map. However, even with these inaccuracies, I expected better results. Let's examine the results using the manually calculated FCD map:

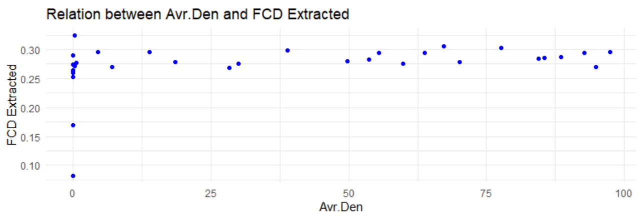



Figure 20: FCD (Manual) values as a function of average densiometer values

This graph shows a slight improvement. We can see that landscapes without forest (Den = 0) generally have different and typically lower values than those with forest. However, we still cannot observe any clear trend for different canopy densities. I believe that a meaningful relationship could be found with a correctly calculated FCD map.

# 8 Construction of a Solar Panel System

During our fieldwork in Garibaldi Park, we needed to capture multispectral drone imagery. Since we were operating in a remote mountainous area, we required a portable energy source to charge the drone batteries. The most practical solution was to build a solar panel system.

## 8.1 Overview of the Solar Panel System

The solar panel system consists of four main components: the solar panels, a charge controller, a battery, and an inverter. Below is a schematic of the installation:

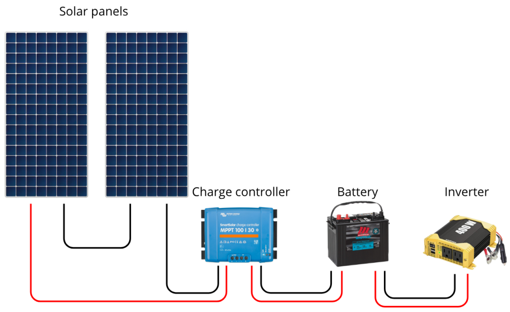



Figure 21: Electrical schematic of the solar panel system

Here is the function of each component:

- **Solar panels**: We used polycrystalline solar panels to convert solar energy into electricity.
- **Charge controller**: This component regulates the battery charging process using energy from the solar panels. In our setup, we used an MPPT (Maximum Power Point Tracking) controller, which employs algorithms to optimize charging efficiency. It also helps protect the battery's health.
- **Battery**: Stores the electricity generated by the solar panels for later use.
- **Inverter**: Converts 12V DC power from the battery into 110V AC power, enabling the use of standard domestic devices.

## 8.2 System Sizing

In this section, we describe how we sized each component and explain our choices.

First, we considered the power requirements: the drone battery charger consumes approximately 100 watts. Assuming it runs for 10 hours a day, this equates to an energy requirement of 1 kWh per day.

- **Solar panels**: We used an online solar panel calculator to estimate the required panel size. Assuming 7 hours of sunlight per day and an environmental efficiency factor of 80% (to account for dust, weather, etc.), the calculator recommended a 180-watt panel. To ensure a safety margin—especially for consecutive cloudy days—we opted for two 150-watt panels, totaling 300 watts. We connected the panels in series to increase the voltage, as MPPT controllers perform better with higher input voltage.

Let's now look at the characteristics of the solar panel system. Each panel provides 150 watts and a maximum current of 8.7 amps. Under ideal conditions, the voltage is around 17V and can rise up to 22V. Since the panels are connected in series, the total voltage doubles (up to 44V), while the current remains at 8.7 amps, and the total power is 300 watts.

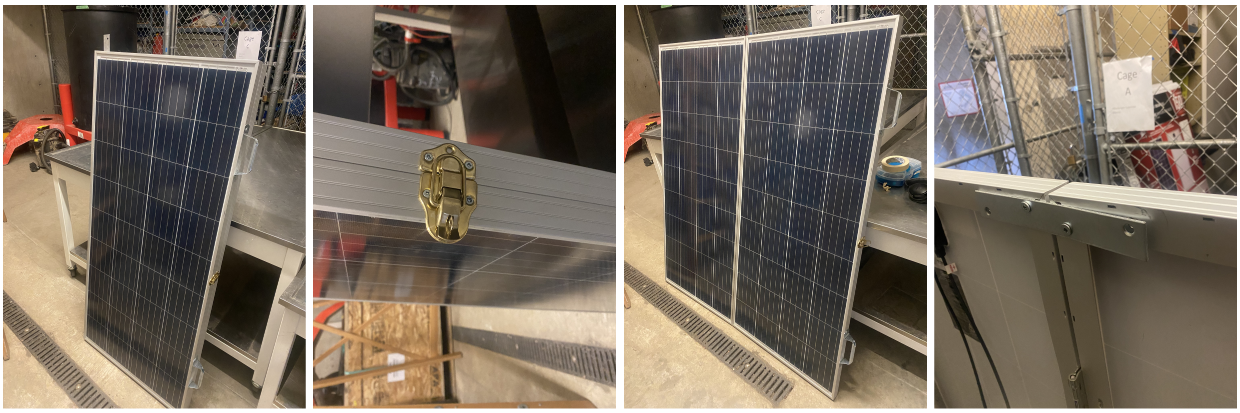
- **Charge controller**: To handle the current (8.7A) and voltage (up to 44V), we selected a 100V / 30A MPPT charge controller. In hindsight, a 75V / 15A controller would have been sufficient.
- **Battery**: We chose a 12V deep-cycle marine battery, which is well-suited for solar systems due to its charge/discharge characteristics. The battery has a capacity of 65 amp-hours, equivalent to 780 Wh.
- **Inverter**: Since the charger consumes 100W, a 400W inverter was sufficient. This also gave us the flexibility to power other small devices if needed.

For the wiring, we used 10 AWG copper cables, which can safely carry up to 30 amps—more than enough for our maximum current. Additionally, we installed two 15-amp fuses: one between the solar panels and the charge controller, and another between the battery and the inverter. These fuses help protect the system components against short circuits or overcurrent situations.

## 8.3 System Construction

In this section, we explain how the solar panel system was assembled. Let's begin with the two solar panels.

To keep the system lightweight and affordable, we designed a foldable structure for the solar panels, allowing for easy storage and transport. We connected the two panels using metal hinges so that they can be folded together. When unfolded, a flat metal bar is used to lock them in place. When folded, a drawer catch lock keeps the structure closed. Two handles were also added for easy carrying.



(a) Solar panel fold (b) Drawer catch lock (c) Solar panel unfold (d) Matel bar

Figure 22: Solar panels structure

The two solar panels are directly connected in series using wires. At the ends of the wires, we attached clamps to easily connect them to the rest of the circuit.

All the remaining components are housed in a single box. On the outside of the box, we mounted two terminals where the panel clamps can be connected. These terminals conduct current into the box.

Inside the box:

- The charge controller is mounted vertically to allow proper air circulation for cooling.
- The controller is wired to the battery, which is then connected to the inverter.
- Both the battery and inverter are securely attached to a wooden base inside the box to keep them in place during transport.

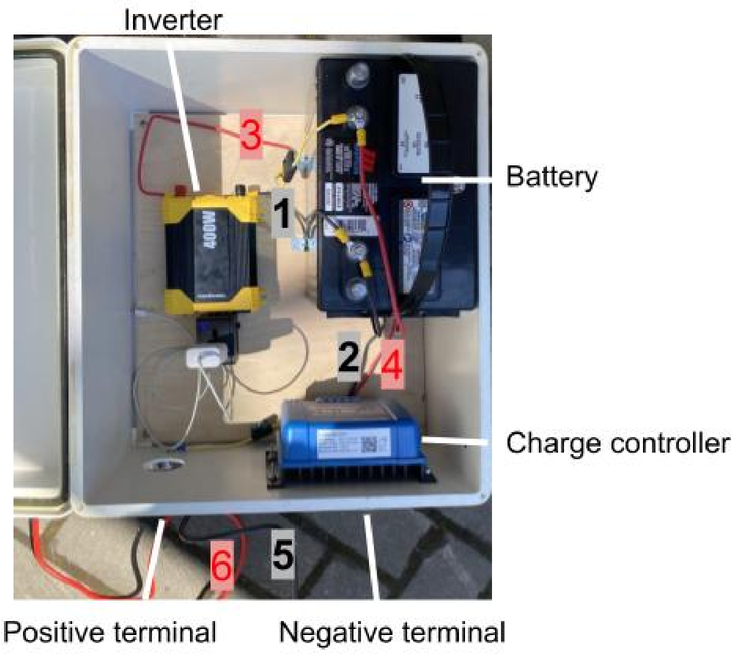



Figure 23: Box with components

## 8.4 System Usage

Here are the instructions for setting up the solar panel, there are working with the schema 23 to identify the wire. They are also numerate in the box in real life.

**Before going into fieldwork:** Download the VictronConnect application on your smartphone or computer.

**Solar Panel Setup Instructions:**

Attention: You can screw and unscrew the nuts by hand, no risk of getting a shock but don't touch the positive and negative terminal at the same time!

1. Connect the black negative wire 1 from the inverter to the negative terminal of the battery.
2. Connect the black negative wire 2 from the charge controller to the negative terminal of the battery.

3. Connect the red/yellow positive wires with the fuse attached 3 from the inverter to the positive terminal of the battery.
4. Connect the positive red wire 4 from the inverter to the positive terminal of the battery.
5. Open the VictronConnect app on your smartphone/computer. You can connect to the charge controller via Bluetooth.
6. Connect the black negative wire 5 from the solar panel to the negative terminal located on the outside of the box.
7. Connect the red positive wire 6 from the solar panel to the positive terminal located on the outside of the box.

Once connected, the application interface will display data similar to the example below. Naturally, the exact values will depend on sunlight conditions:


Figure 24: VictronConnect application interface

In the screenshot, we can observe that the solar panel is producing 239 watts, with a voltage of 34.29V and a current of 7A. These values are consistent with the specifications discussed earlier. The battery shows a voltage of 13.78V and a current of 16.2A.

The charge controller can be in one of three states, which you can see either in the app or directly on the charge controller via LEDs.

- **Bulk**: The battery is being charged at maximum current.
- **Absorption**: The battery is nearly full (typically over 80%). The charge controller gradually reduces the current based on the battery's charge level.
- **Float**: The battery is fully charged. The controller maintains a low current to prevent self-discharge.

To charge any device, simply plug it into the inverter and turn it on using the button next to the outlet. Keep the inverter turned off when it is not in use.

# 9 Field Work

For taking some data in the park I had to organize an field work trip to Garibaldi park. The idea is to estimise the canopy cover and the trees diensity in different points in the park.

## 9.1 Preparation of the field work

The first step was to preparing the field work. The idea was to cover different zone in the park, for that we have separated the trip in three trip of three days for a total of ten days:

1. First one in Helm creek
2. Second one in Efin lake, Rampart ponds, south of park
3. Third one in Russet lake

Here is the map with the three trails.

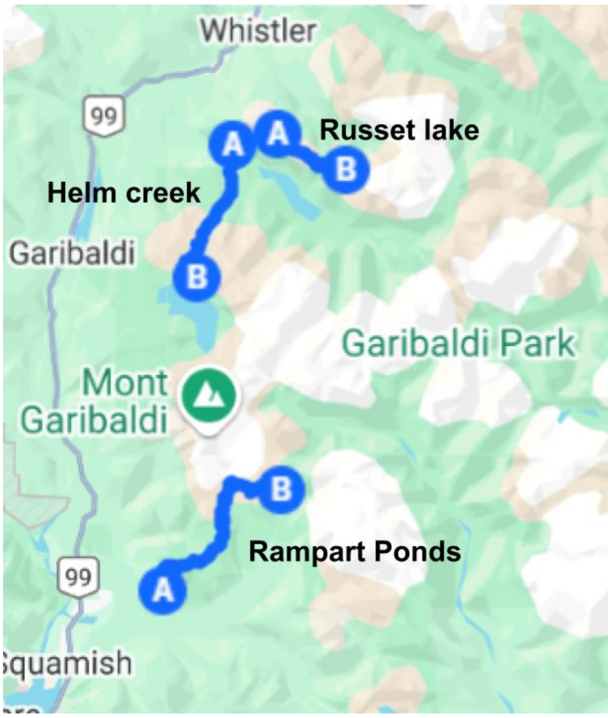



Figure 25: Field work trails

With this three trips we cover only the west of the park which is not the best cover for the data, but there are also the most easy way to go to in the park. The other part of the park are really deep nature so way more difficult to access and stay.

We had to prepare all the camping stuff, dehydrate food and also plan all the logistics to going to the site.

The all these trip we have plan in advance couple of points. I have made it manually by selecting them with satellite view. I separe the point in three differents groups:

- Meadows (Bleu)
- Low density forest (In Yellow)

- High density forest (Red)

I tried to took the same amount of point in each categories. With the retreats, I will not manually take this point but randomly in a range on the side of the trail. Because by manually taking the point I included a biais which can have an influence on the results.

Here the map with all the planned point:

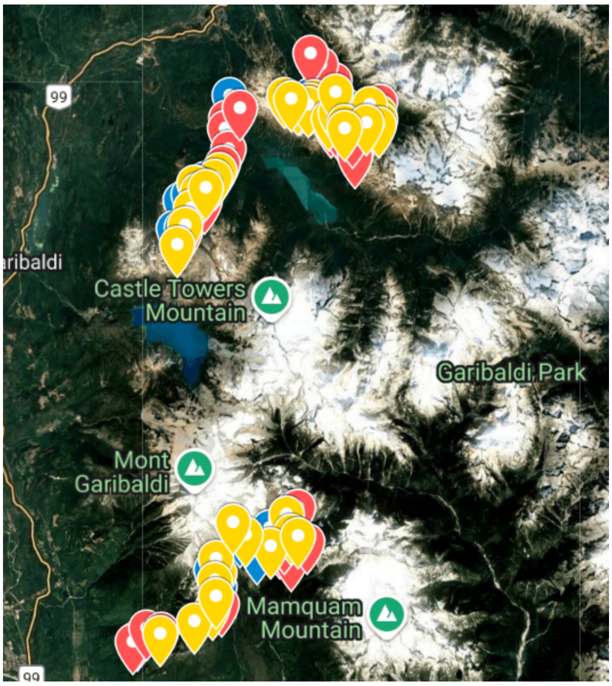



Figure 26: Field work points

## 9.2 Point taking

The main goal for each point was to take canopy and tree density, we have also took the spieces of trees we find and the type of soil.

Let us first see how to use a densiometer, this is a tool to measure the tree canopy.

Figure 27: Spherical Densiometer

A densiometer is a tool with a spherical mirror. This mirror have a grid on it. Here is a schema of this grid:

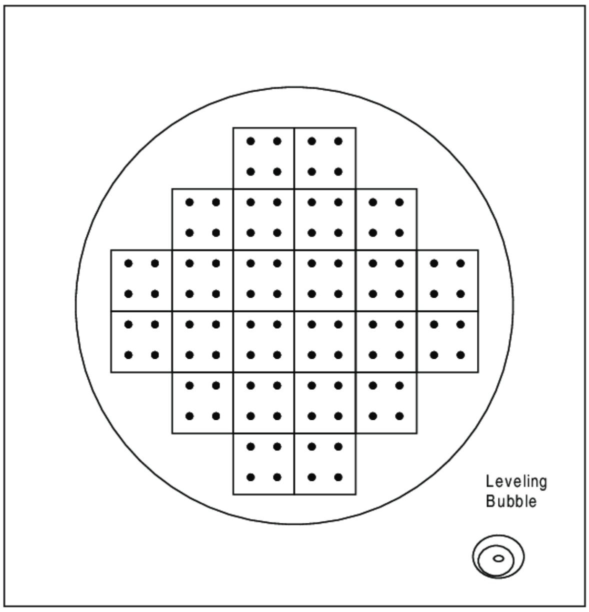



Figure 28: Schema grid densiometer

To estimate the canopy density, we need to count the numbers of points who are cover/uncover by canopy. We are doing this for the for cardinal direction, to avoid the biais of the slope, and we averaging them. Then to have a result in percentage, we can multiply the results by $100/96$.

For the tree density we had use a technique who consist mesuring the distance between the point and the 5th and 10th tree. This distance was took with a mesure tape or a ranger finder when the trees was to far. With this mesures we can calcule the tree density in a circle of radius of the mesure. For exemple if the 10th tree is at 5 meter: we have 10 trees in a 5 meter radius circle. We have to count the 10th tree as a 0.5 tree because we count the distance from the middle of the tronc. We have:

$$D_{trees} = \frac{9.5}{\pi * 5^2} \approx 0.1210\, trees.m^{-2} \tag{2}$$

## 9.3 Reality of field work

In reality the field was a bit frustrating in term of work. Firstly we had to cancel the frist trip to Helm Creek because of Cougars in the area. Then taking points was, least for the firsts ones, way more long than expecting. So I couldn't took as much point than I expected. Moreover I think that I have plan to big hikes for having clearly the time to work during then.

Howerver, I could took in total 31 points during the two trip of 3 days each. In additions, during this days I could see in real the field I'm mapping, this give me an other point of view than the satellite image. This makes me realize some things, and make me think about the result I want, and how to have it.

## 9.4 Last trip with UBC (University of British Columbia)

I have made one last trip with UBC teams that also work with us in Garibaldi Park. The goal of this trip was to took some drones images the some trails in Taylor meadow for studying, the trampling effect around the trails on plant. On the trip I have use a DGPS (Differential Global Positioning System).

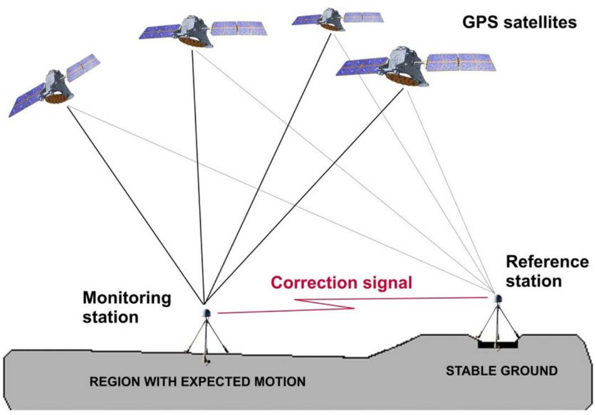



Figure 29: Schema Differential Global Positioning System

This tool is made to take very precise relative GPS point. You have to set up a base (Reference station) on a knew point or average the GPS position for long time (I used 20 minutes). After that you use de Rover (Monitoring station) to take your GPS point. We this tool you have a really high precision of position between the base and the rover (less than a centimetre).

I used this tool to obtain the coordinates of some target. This target is then the coordinate reference in the drone images. Then with some images post-processing, we can extend the coordinate to all pixels.

# 10 Conclusion

To conclude this report, I can say that this internship taught me a great deal on multiple levels. First, I learned how to conduct scientific documentation and how to extract relevant information from scientific papers. It also allowed me to make many mistakes, which I will avoid in the future. For instance, I realize now that I moved too quickly into image manipulation without first gaining a comprehensive understanding of the subject. This cost me a significant amount of time. In the future, I will not hesitate to spend more time on documentation and planning before diving into practical work.

This internship was somewhat frustrating for me because, in just five months, I felt that I did not have enough time to fully grasp the subject or achieve meaningful results. It often seemed like I spent most of my time trying to figure out how to do things rather than actually producing results. However, I understand that this is part of the research process. Now, I have a solid foundation in alpine forests and image processing, and I know where to find information and how to use it effectively.

This experience was also an opportunity for me to significantly improve my English. Even though my supervisor was a French speaker, I spent almost all my time speaking English, both at work and in my personal life. While I still don't consider myself fluent, my English skills have improved greatly. Additionally, this internship gave me the chance to discover the beautiful province of British Columbia.

During this period, I also learned a lot about plants, trees, and nature, both through fieldwork and my research.

This internship has made me want to continue in the world of research. I really appreciated the freedom that comes with conducting research, even if it can sometimes be challenging to achieve good results. I also believe that research is a meaningful way to contribute to the world. However, I don't think I will continue specifically in image processing, as it was not the most exciting aspect of IT work I have experienced.

Finally, I would like to express my gratitude once again to Noémie Boulanger-Lapointe for hosting me, as well as to UVic and Seatech, without whom this experience would not have been possible. I also want to thank Audrey Minghelly for her guidance and for reviewing this lengthy report.

# Bibliography


Jonathan A. Flores, Colin J. Gleason, Craig B. Brinkerhoff, Merritt E. Harlan, M. Malisse Lummus, Leigh A. Stearns, and Dongmei Feng. Mapping proglacial headwater streams in High Mountain Asia using PlanetScope imagery. *Remote Sensing of Environment*, 306:114124, May 2024. ISSN 00344257. doi: 10.1016/j.rse.2024.114124. URL https://linkinghub.elsevier.com/retrieve/pii/S0034425724001354.

Zhongbin Li, Hankui K. Zhang, David P. Roy, Lin Yan, and Haiyan Huang. Sharpening the Sentinel-2 10 and 20 m Bands to Planetscope-0 3 m Resolution. *Remote Sensing*, 12(15):2406, July 2020. ISSN 2072-4292. doi: 10.3390/rs12152406. URL https://www.mdpi.com/2072-4292/12/15/2406.

Heather Reese, Mattias Nyström, Karin Nordkvist, and Håkan Olsson. Combining airborne laser scanning data and optical satellite data for classification of alpine vegetation. *International Journal of Applied Earth Observation and Geoinformation*, 27:81–90, April 2014. ISSN 15698432. doi: 10.1016/j.jag.2013.05.003. URL https://linkinghub.elsevier.com/retrieve/pii/S0303243413000524.

Gang Wu, Chunhong Pan, V. Prinnet, and Wei Liu. Comparative research on landuse classification by fusing optical image and DEM. In *Proceedings. 2005 IEEE International Geoscience and Remote Sensing Symposium, 2005. IGARSS '05.*, volume 6, pages 3928–3930, July 2005. doi: 10.1109/IGARSS.2005.1525770. URL https://ieeexplore.ieee.org/document/1525770/. ISSN: 2153-7003.

Pegah Mohammadpour, Domingos Xavier Viegas, and Carlos Viegas. Vegetation Mapping with Random Forest Using Sentinel 2 and GLCM Texture Feature—A Case Study for Lousã Region, Portugal. *Remote Sensing*, 14(18):4585, September 2022. ISSN 2072-4292. doi: 10.3390/rs14184585. URL https://www.mdpi.com/2072-4292/14/18/4585.

Training data in satellite image classification for land cover mapping: a review - University of Victoria. URL https://search.library.uvic.ca.

Classification of Alpine Grasslands in Cold and High Altitudes Based on Multispectral Landsat-8 Images: A Case Study in Sanjiangyuan National Park, China - University of Victoria. URL https://search.library.uvic.ca.

Azar Zafari, Raul Zurita-Milla, and Emma Izquierdo-Verdiguier. Evaluating the Performance of a Random Forest Kernel for Land Cover Classification. *Remote Sensing*, 11(5):575, March 2019. ISSN 2072-4292. doi: 10.3390/rs11050575. URL https://www.mdpi.com/2072-4292/11/5/575.

Mehebub Sahana, Haroon Sajjad, and Raihan Ahmed. Assessing spatio-temporal health of forest cover using forest canopy density model and forest fragmentation approach in Sundarban reserve forest, India. *Modeling Earth Systems and Environment*, 1(4):49, December 2015.

ISSN 2363-6211. doi: 10.1007/s40808-015-0043-0. URL https://doi.org/10.1007/s40808-015-0043-0.

FOREST CANOPY DENSITY ESTIMATING, USING SATELLITE IMAGES. 2008.

# 1 Annex

**Annex A: Compute canada tutorial**

# Tutorial Compute Canada

This document is designed to help you get started with Compute Canada. It covers how to connect to the system, manage your files, and run code efficiently. Most of the information presented here is based on the official Compute Canada documentation. I also recommend checking out this excellent tutorial, which provides a clear and practical overview.

**Author:** Côme Beauquier beauquiercome@gmail.com

## 2 Connect to Compute Canada (SSH Key)

The first step is to connect your computer to a Compute Canada server. To do this, we will use an SSH key. I recommend watching the following video, which clearly explains how to use an SSH key to connect your computer to Compute Canada. If the video doesn't work, don't worry. Let's go through the steps together.

1. **Open a command terminal**. For example, use PowerShell on Windows, or Terminal on macOS/Linux. For remember, here the common terminal commands.

2. **Create an SSH key** with the following command:

   ```
   ssh-keygen -t ed25519
   ```

   This command will generate two files in a folder named /.ssh:

   - your private key (used for authentication)
   - your public key (used to authorize access)

3. **Get your public key**. You can open the '.pub' file in a text editor, or use the following command in your terminal (in the folder /.ssh):

   ```
   cat yourfile.pub
   ```

   Copy the entire public key (it will start with 'ssh-ed25519').

4. **Paste your key into Compute Canada**. Go to *My Account > Manage SSH Keys* on the Compute Canada website, and paste the public key into the form.

5. **Connect your terminal to a Compute Canada server**. Compute Canada has five main servers: Béluga, Narval, Cedar, Graham, and Niagara. Use the following command to connect using your username and the name of the chosen server:

   ```
   ssh username@servername.computecanada.ca
   ```

   If the above command doesn't work, you can try connecting by explicitly providing the path to your private key:

   ```
   ssh -i path/to/privatekey username@servername.computecanada.ca
   ```

You are now connected to Compute Canada through your terminal, congratulations! I also recommend enrolling your Compute Canada account in DUO (multi-factor authentication). You can do this under *My Account > Multifactor Authentication Management*. Without this, you may encounter login issues in the future.

Once you are connected, you will see that there are three directories in your home folder:

- **scratch**: A high-performance working space, ideal for running jobs. However, data stored here will be deleted after 60 days.
- **projects**: Used to store and share data with collaborators.
- **nearline**: Intended for long-term data storage.

You can also create new folders in your home directory for small personal files, such as scripts.

# 3 Managing Files on the Compute Canada Server

There are two main ways to manage files on Compute Canada: using the terminal, or using software such as Globus or WinSCP. Let's explore both methods.

## 3.1 Using the Terminal

To transfer files between your local computer and the server, you can use the `scp` (secure copy) command.

1. **Open a terminal** without connecting to Compute Canada.
2. **Download a file or folder** from Compute Canada using:

```
scp username@servername.computecanada.ca:path/to/fileOrFolder
"C:\path\to\yourLocalStorage"
```

3. **Upload a file or folder** to Compute Canada using:

```
scp "C:\path\to\yourFileOrFolder"
username@servername.computecanada.ca:path/to/remoteStorage
```

**Note:**

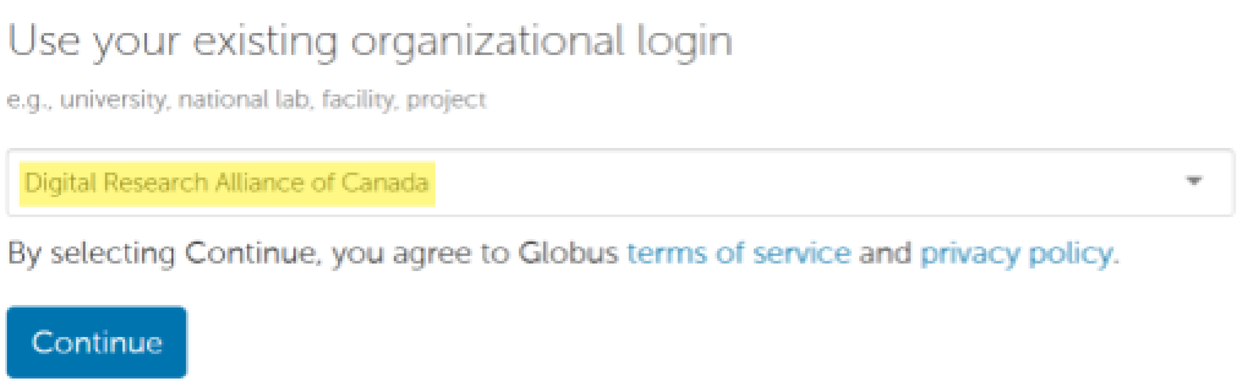
- The `scp` command may not work properly in some terminals such as PowerShell. I personally used Git Bash, which works well.
- The path on Compute Canada should start from the child directory inside your home folder. Here's an example:

```
$ scp -r come@cedar.computecanada.ca:projects/def-nbl/come/codes "C:\Users\beauq
\Desktop\Test"
```

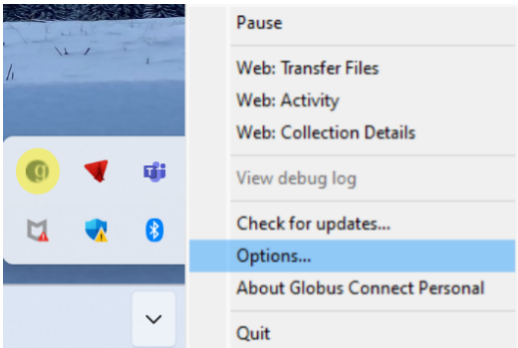
## 3.2 Using a software

In this example, we'll use Globus, but the process should be similar for WinSCP or other software.

1. **Set up your Globus account** here, using your Compute Canada login, and **download Globus Connect Personal**.

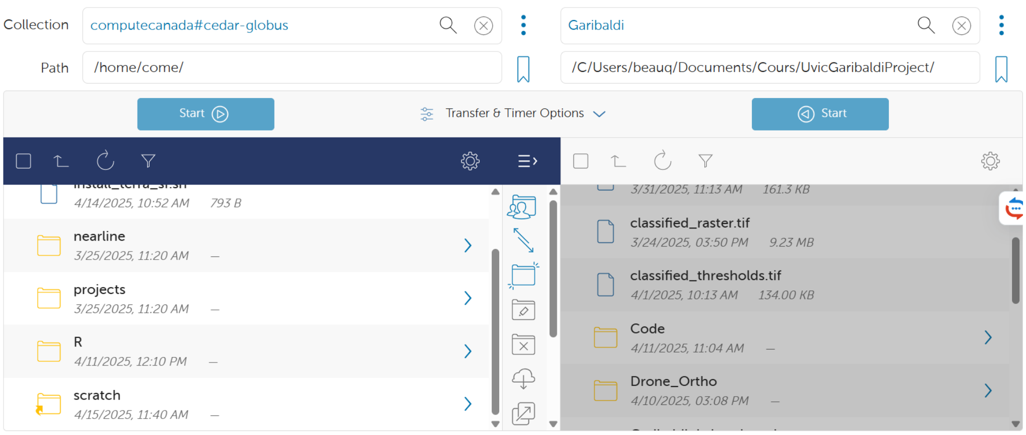


2. **Connect your working folders**: By default, Globus Connect Personal is linked to your `Documents` folder. You can add additional folders by right-clicking the Globus icon and selecting "Options":




   Then, in the options menu, click the "+" button to add the folder(s) you want to use.

3. **Transfer your data**: Select your Compute Canada server and your local folder. Choose the files or folders to transfer and click "Start" to begin the transfer.




# 4 Running R Code on Compute Canada

Before running R code on Compute Canada, let's first go over how to install R packages.

## 4.1 Installing Packages

1. **Open a terminal** and connect to a Compute Canada server using:

```
ssh username@servername.computecanada.ca
```

2. **Start an R session** with:

```
R
```

   Choose the R module (version) you want to work with.

3. **Install your package** using:

```
install.packages("yourPackage", repos =
"https://cloud.r-project.org")
```

This works for most standard packages, but not all. For example, I encountered issues when trying to install the 'terra' package.

Let's now look at how I installed the 'terra' package — this process can also be applied to other complex packages.

## 4.2 Creating a Job Script

To run R code on Compute Canada, we need to create a job script that instructs the server how to run the script. This script is written in Bash.

- **Open a terminal and connect** to Compute Canada.
- **Create a folder** to store your job script, for example in your 'projects' or 'scratch' directory:

```
mkdir my_job
```

- **Create the job script file**:

```
nano job_script.sh
```

This opens a text editor where you can write your script.

- **Write a basic job script:**

```
#!/bin/bash
#SBATCH --time=00:15:00
#SBATCH --account=def-user
echo 'hello world'
```

Here's what each line does:

  – '–time': Sets the maximum job duration.
  – '–account': Specifies your project ID (e.g., 'def-nbl') so Compute Canada knows which project the job belongs to.

Save the file with 'Ctrl + S' and exit with 'Ctrl + X'.

- **Submit your job:**

```
sbatch job_script.sh
```

- **Monitor your job:**

```
squeue -u yourUsername
```

When the job finishes, an '.out' file will appear in your folder. You can view it using:

```
cat yourFile.out
```

You should see the message "hello world."

- **Example of a more advanced job script:**

```
#!/bin/bash
#SBATCH --time=01:00:00
#SBATCH --account=def-user
#SBATCH --cpus-per-task=4
#SBATCH --mem=16G
#SBATCH --output=log_%j.txt
#SBATCH --error=error_%j.txt
```

Additional options:

  - '–cpus-per-task': Number of CPUs.
  - '–mem': RAM allocation.
  - '–output': Output log file.
  - '–error': Error log file.

For more information, visit the official documentation.

## 4.3 Running an R Script

In the '.sh' file, load the required R module and any dependencies. For example, to load R version 4.2.2:

```
module load StdEnv/2020 r/4.2.2
```

Then run your R script with:

```
Rscript /path/to/yourscript.r
```

Note: As with the 'scp' command, paths should start after your home directory. For example:

```
Rscript /project/def-nbl/come/codes/test.R
```

Also be careful when accessing paths inside Compute Canada: use '/project/' not '/projects/'.

You can now save and launch the job script as shown before.

## 4.4 Installing the ‘terra‘ Package

To install the ‘terra‘ package, I first changed the repository source. By default, R uses ‘https://cloud.r-project.org‘, which is fast and stable, but it doesn’t always work with complex packages like ‘terra‘.

Instead, use:

```
install.packages("terra", repos =
"https://mirror.csclub.uwaterloo.ca/CRAN/",
dependencies = TRUE)
```

Although this works, it can be very slow on personal computers. To make installation faster and more reliable, I ran it on Compute Canada using the following job script:

```
#!/bin/bash
#SBATCH --account=def-nbl
#SBATCH --job-name=install_r_packages
#SBATCH --output=install_r_packages_%j.log
#SBATCH --time=01:00:00
#SBATCH --cpus-per-task=4
#SBATCH --mem=16G

# Load required modules
module load StdEnv/2020 gcc/9.3.0 udunits/2.2.28 \
gdal/3.5.1 r/4.2.2

# Create a folder to store the packages
mkdir -p $HOME/R/x86_64-pc-linux-gnu-library/4.2
export R_LIBS="$HOME/R/x86_64-pc-linux-gnu-library/4.2:$R_LIBS"

# Install packages
R -e "install.packages(c('terra', 'sf'),
repos='https://mirror.csclub.uwaterloo.ca/CRAN/',
dependencies=TRUE)"
```

This script requests 4 CPUs and 16 GB of RAM. The required modules are:

- **StdEnv**: Standard environment.
- **gcc**: C/C++ compiler.
- **udunits**: Unit handling library.
- **gdal**: Library for geospatial data.
- **r**: The R language itself.

Once installation completes, you can verify it in an R session by running:

```
library(terra)
```