Lichen Abundance Detection in Antarctica using Machine Learning

By

Harsh Joshi (ID No. 18CEUON129)

Dhruvkumar Kakadiya (ID No. 18CEUOS032)

In

partial fulfillment of the requirements

For the degree of

BACHELOR OF TECHNOLOGY

In

Computer Engineering

Internal Guide
Prof. Malay S. Bhatt
Associate Professor
Dept. of Comp. Engg.

External Guide
Dr. Chandra Prakash Singh
(Scientist)
Space Application Center (ISRO)

Faculty of Technology
Department of Computer Engineering
Dharmsinh Desai University
April 2022

CERTIFICATE

This is to certify that the project work titled

Lichen Abundance Detection in Antarctica
using Machine Learning

is the bonafide work of

Harsh Joshi (ID No. 18CEUON129)
Dhruvkumar Kakadiya (ID No. 18CEUOS032)

Carried out in the partial fulfillment of the degree of Bachelor of Technology in

Computer Engineering at Dharmsinh Desai University
in the academic session
December 2021 to April 2022.

Prof. Malay S. Bhatt
Associate professor
Dept. of Comp. Engg.

Dr. C. K. Bhensdadia
(Head)
Dept. of Comp. Engg.

Faculty of Technology
Department of Computer Engineering
Dharmsinh Desai University
April 2022

Acknowledgements

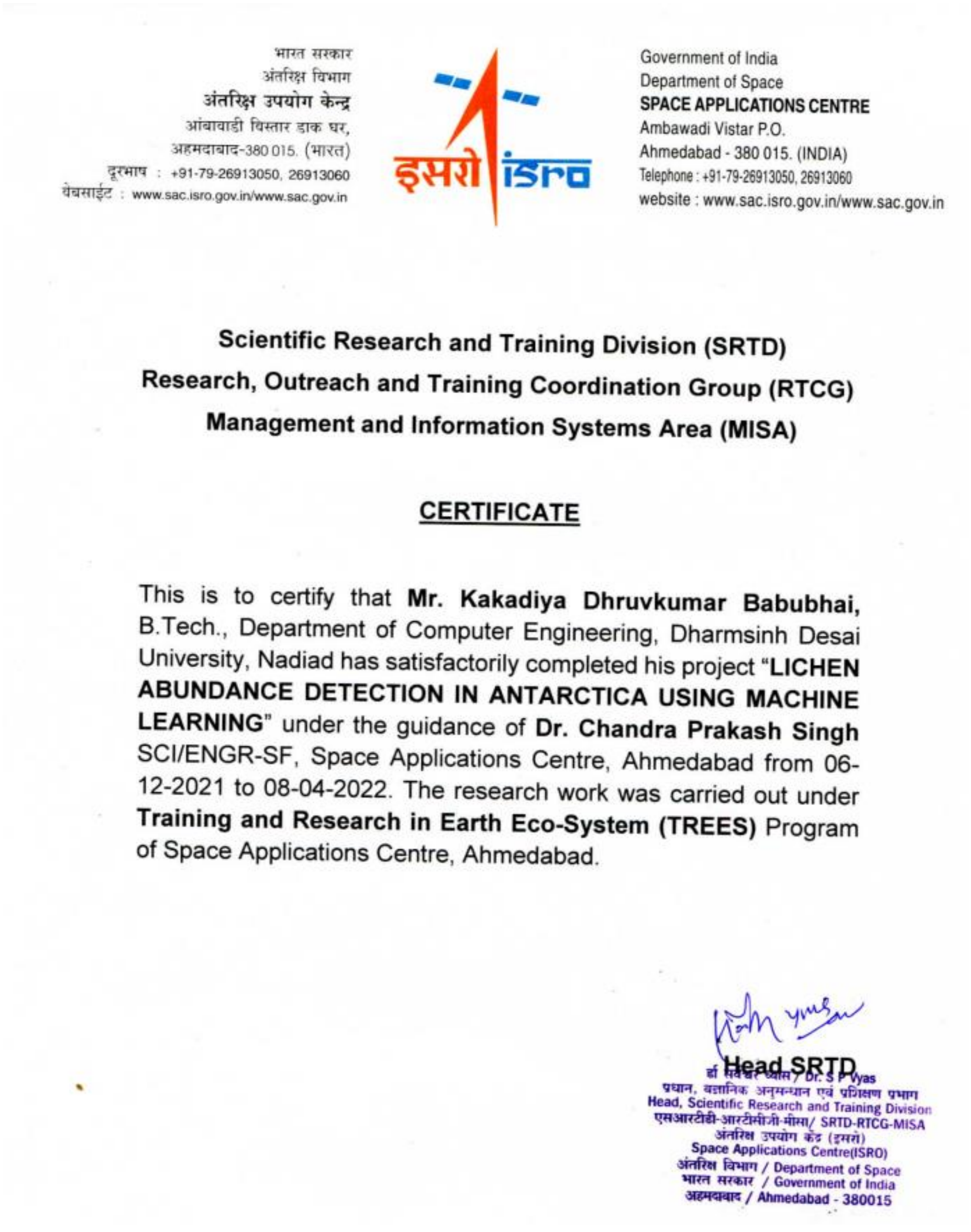
We would like to express our greatest appreciation to all individuals who have helped and supported us throughout the course of this study.

We take this opportunity to express our profound gratitude and deep regards to our guide **Prof. Malay S. Bhatt** (Professor, CE Dept., DDIT) for his exemplary guidance, monitoring, and constant encouragement throughout the course of this internship.

It is our radiant sentiment to place on record our best regards, deepest sense of gratitude to **Dr. C. P. Singh** (Scientist, Space Application Center, ISRO) for his precious guidance, which was extremely valuable for our study both theoretically and practically.

We also take this opportunity to express a deep sense of gratitude to **Dr. C. K. Bhensdadia** (Head of Department, CE Dept., DDIT). We consider ourselves fortunate individuals for being provided with such an enriching opportunity to interact with and learn from highly intellectual professionals.

We perceive this opportunity as a big milestone in our career development ans we will strive to use gained skills and knowledge in the best possible way.



भारत सरकार
अंतरिक्ष विभाग
अंतरिक्ष उपयोग केन्द्र
आंबावाडी विस्तार डाक घर,
अहमदाबाद-380 015. (भारत)
दूरभाष : +91-79-26913050, 26913060
वेबसाईट : www.sac.isro.gov.in/www.sac.gov.in

Government of India
Department of Space
SPACE APPLICATIONS CENTRE
Ambawadi Vistar P.O.
Ahmedabad - 380 015. (INDIA)
Telephone : +91-79-26913050, 26913060
website : www.sac.isro.gov.in/www.sac.gov.in

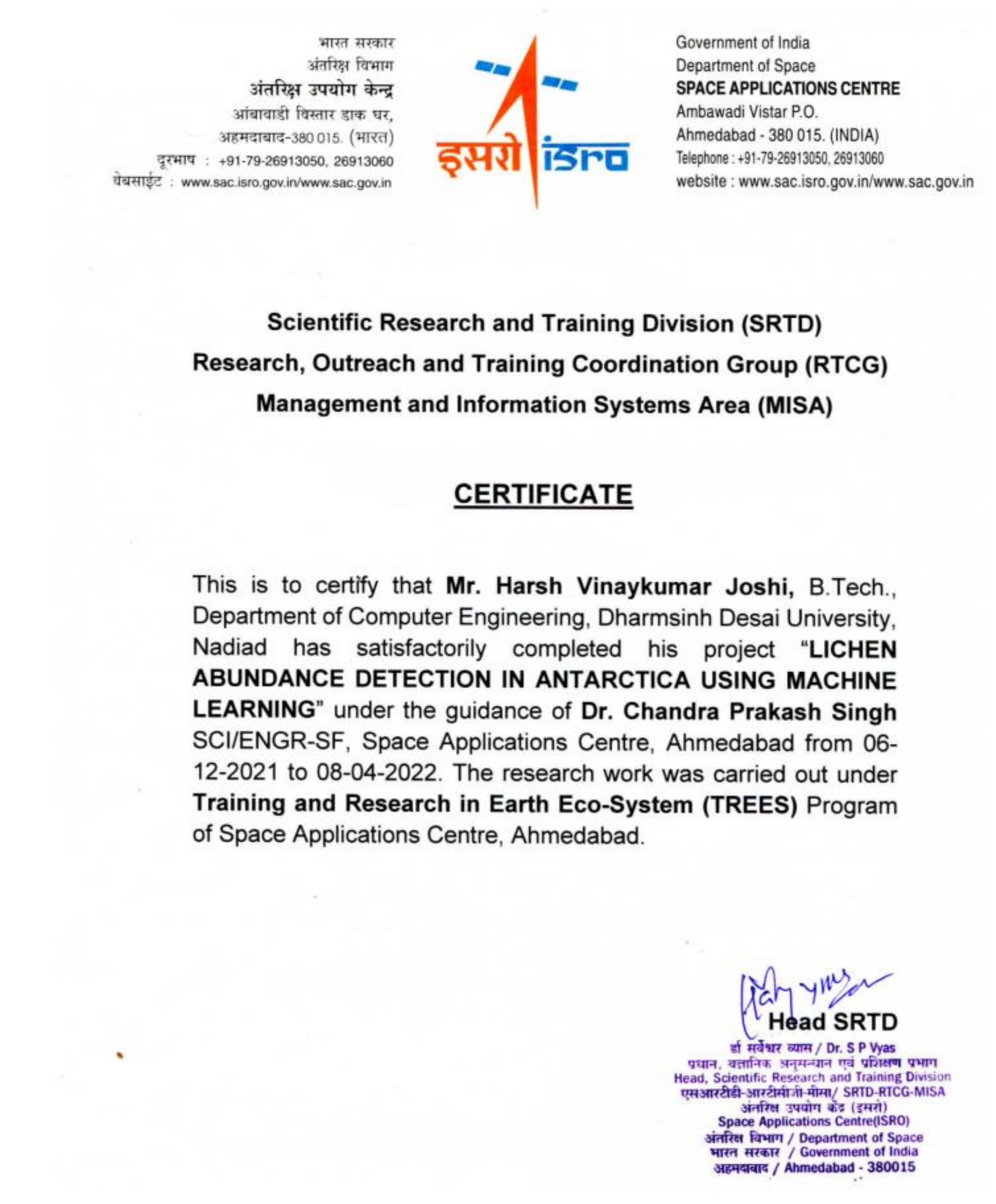
Scientific Research and Training Division (SRTD)
Research, Outreach and Training Coordination Group (RTCG)
Management and Information Systems Area (MISA)

CERTIFICATE

This is to certify that **Mr. Kakadiya Dhruvkumar Babubhai,** B.Tech., Department of Computer Engineering, Dharmsinh Desai University, Nadiad has satisfactorily completed his project "**LICHEN ABUNDANCE DETECTION IN ANTARCTICA USING MACHINE LEARNING**" under the guidance of **Dr. Chandra Prakash Singh** SCI/ENGR-SF, Space Applications Centre, Ahmedabad from 06-12-2021 to 08-04-2022. The research work was carried out under **Training and Research in Earth Eco-System (TREES)** Program of Space Applications Centre, Ahmedabad.

Head SRTD

डॉ [illegible] व्यास / Dr. S P Vyas
प्रधान, वैज्ञानिक अनुसन्धान एवं प्रशिक्षण प्रभाग
Head, Scientific Research and Training Division
एसआरटीडी-आरटीसीजी-मीसा/ SRTD-RTCG-MISA
अंतरिक्ष उपयोग केंद्र (इसरो)
Space Applications Centre(ISRO)
अंतरिक्ष विभाग / Department of Space
भारत सरकार / Government of India
अहमदाबाद / Ahmedabad - 380015



भारत सरकार
अंतरिक्ष विभाग
अंतरिक्ष उपयोग केन्द्र
आंबावाडी विस्तार डाक घर,
अहमदाबाद-380 015. (भारत)
दूरभाष : +91-79-26913050, 26913060
वेबसाईट : www.sac.isro.gov.in/www.sac.gov.in

Government of India
Department of Space
SPACE APPLICATIONS CENTRE
Ambawadi Vistar P.O.
Ahmedabad - 380 015. (INDIA)
Telephone : +91-79-26913050, 26913060
website : www.sac.isro.gov.in/www.sac.gov.in

Scientific Research and Training Division (SRTD)
Research, Outreach and Training Coordination Group (RTCG)
Management and Information Systems Area (MISA)

CERTIFICATE

This is to certify that **Mr. Harsh Vinaykumar Joshi,** B.Tech., Department of Computer Engineering, Dharmsinh Desai University, Nadiad has satisfactorily completed his project "**LICHEN ABUNDANCE DETECTION IN ANTARCTICA USING MACHINE LEARNING**" under the guidance of **Dr. Chandra Prakash Singh** SCI/ENGR-SF, Space Applications Centre, Ahmedabad from 06-12-2021 to 08-04-2022. The research work was carried out under **Training and Research in Earth Eco-System (TREES)** Program of Space Applications Centre, Ahmedabad.

Head SRTD

डॉ सर्वेश्वर व्यास / Dr. S P Vyas
प्रधान, वैज्ञानिक अनुसन्धान एवं प्रशिक्षण प्रभाग
Head, Scientific Research and Training Division
एसआरटीडी-आरटीसीजी-मीसा/ SRTD-RTCG-MISA
अंतरिक्ष उपयोग केंद्र (इसरो)
Space Applications Centre(ISRO)
अंतरिक्ष विभाग / Department of Space
भारत सरकार / Government of India
अहमदाबाद / Ahmedabad - 380015

Table of Contents

Figure Table

Chapter 1

Introduction

1.1 Introduction to Satellite

A satellite is an object that orbits the sun, earth, or some other massive body. There are two categories of satellites: Natural and man-made. Ordinary satellites follow an elliptical orbit.

Planets, moons, and comets are examples of natural satellites. Many man-made satellites orbit our planet, and they are used for data gathering and communication. At present, more than 2,500 man-made satellites are circling the earth.

Types of satellites:-

1. **Geostationary satellite:-**

- Around 36,000 km away from the earth's surface.
- They are launched by a GSLV rocket. The speed of these satellites is 3.14 km/second.
- These Satellites as seen from Earth, will be at the same spot throughout as they pivot similarly to the earth. They are utilized as correspondence satellites, and for climate-based applications.

2. **Polar satellite:-**

- Unlike geostationary satellites, polar satellites orbit the earth in a north-south direction.
- They are launched using a PSLV rocket and have a height of 600-900 km.
- The speed of these satellites is 8 km/second and one orbit takes 84 minutes.

  They are extremely helpful in applications where the field vision of the whole earth is needed in a single day.

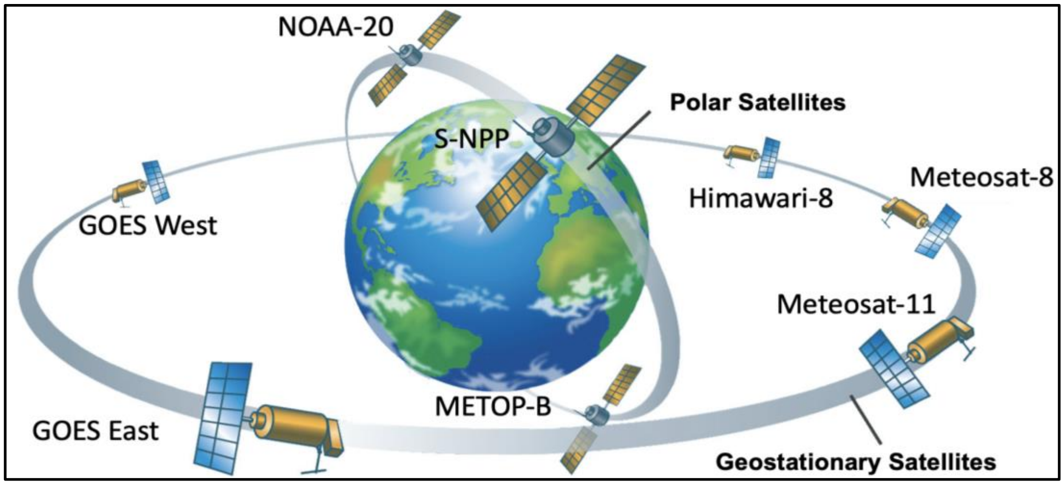


Fig 1.1 Geostationary and Polar satellites orbits and examples

1.2 Introduction to Remote Sensing

Remote sensing, researchers can gain information about Earth's surface without actually touching it. The following image depicts various stages of remote sensing,

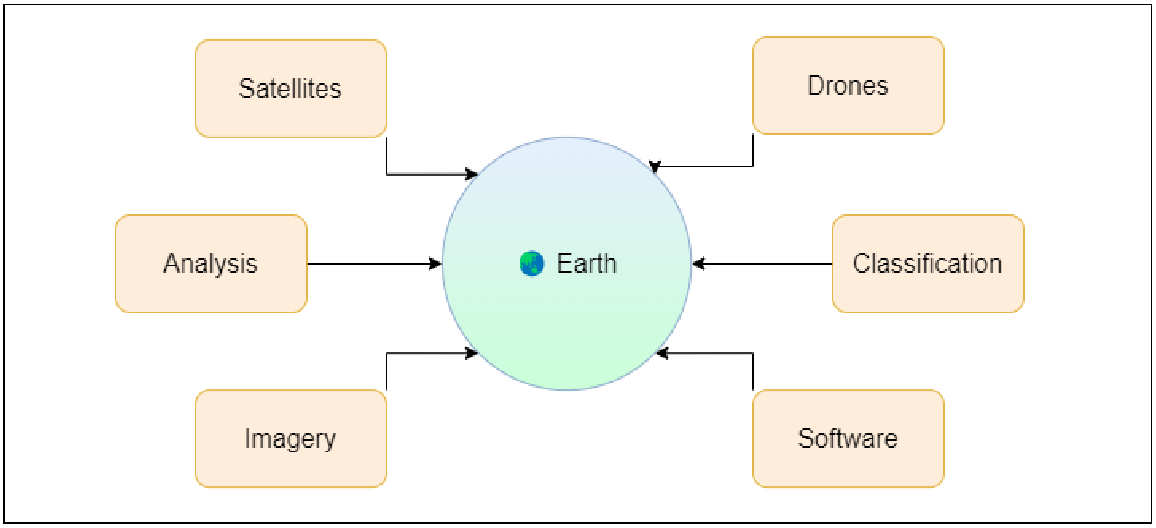


Fig 1.2 Stages of remote sensing

1.3 Lichen

A lichen is not a single organism. Rather, it is a symbiosis between different organisms - a fungus and an alga or cyanobacterium. Cyanobacteria are sometimes referred to as blue-green algae, though they are quite distinct from the algae. Lichens are non-vascular, small plants that grow on rocks, roofs, and other man-made objects.

Lichens constitute the most diverse part of the terrestrial ecosystem on the Antarctic Peninsula. The Antarctic Peninsula has limited detailed information about lichen distributions, and the results of the available sampling are quite heterogeneous. Lichens may be of varied colors - Grey, Greyish green, yellow, brown, and red.

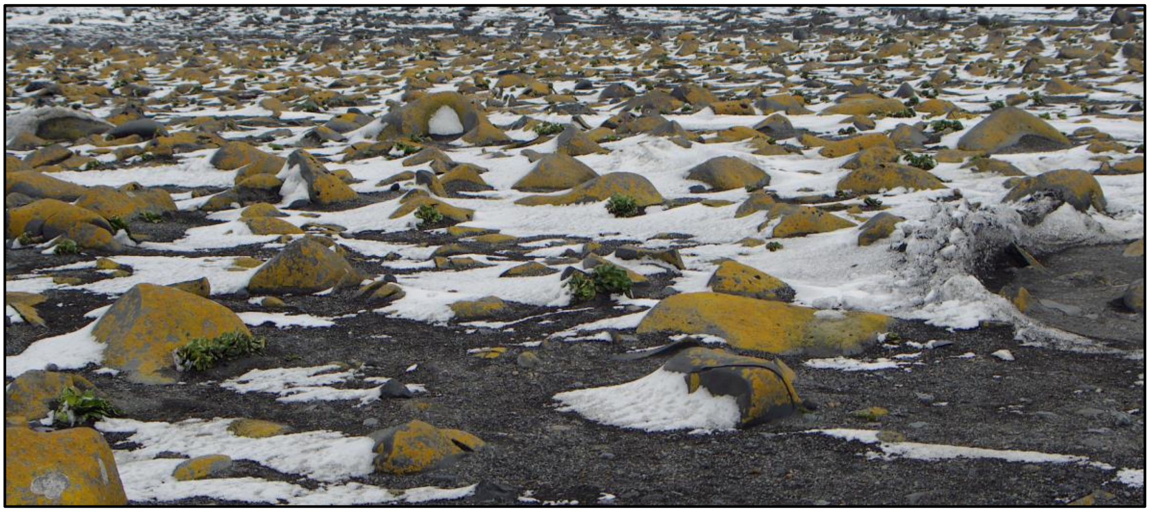
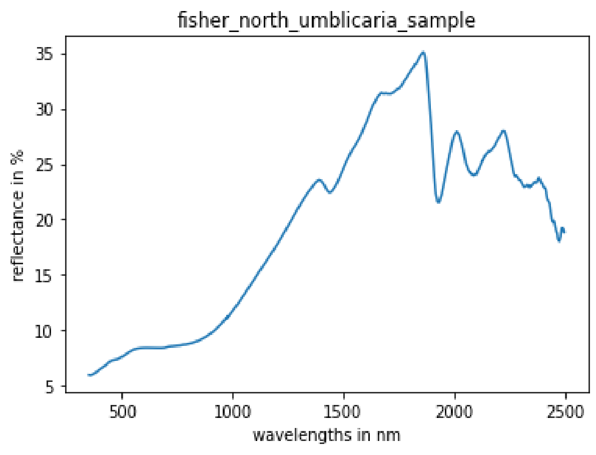


Fig 1.3 Lichens in Australian Antarctica

1.4 Spectral reflectance of Lichen

An Analytical spectral device(ASD) -spectrometer, was used to acquire lichen spectra. The spectrometer captures continuous spectra spanning the 350 nm-2500 nm spectral range. The following illustration depicts a sample visualization used to study the spectral response.



Instrument: SR-3500_SN19480P8
Measurement: Reflectance

Wavelength: 350nm - 2500 nm
Latitude: -69.39072
Longitude: 76.24108

X-axis: wavelength in nm.
Y-axis: reflectance in %

Fig 1.4 Lichen spectra of fisher north Umblicaria

1.5 Google Earth Engine

Google earth engine provides access to the Earth Engine Data Catalog's huge collection of global and regional datasets. It lets users quickly explore data by zooming and panning anywhere on Earth, adjusting display settings, and layering data to examine change over time.

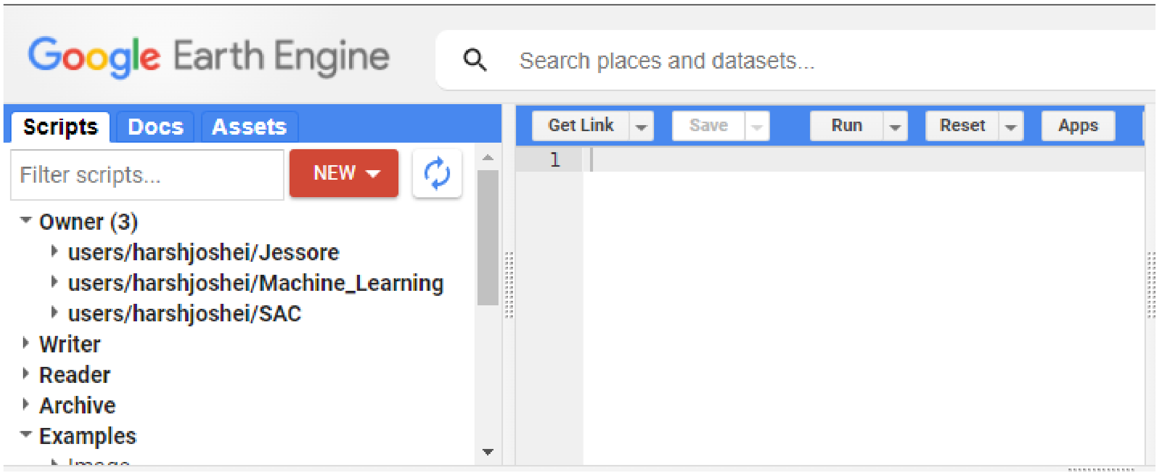


Fig 1.5 google earth engine

Chapter 2

About the System

2.1 Sentinel-2 Multispectral Instrument

The Sentinels are a constellation of satellites developed by ESA to operationalize the Copernicus program, which includes all-weather radar images from Sentinel-1A and 1B, high-resolution optical images from Sentinel-2A and 2B, ocean and land data suitable for environmental and climate monitoring from Sentinel-3, as well as air quality data from Sentinel-5P.

It is a wide-swath, high-resolution, multi-spectral imaging mission supporting Copernicus Land Monitoring studies, including the monitoring of vegetation, soil, and water cover, as well as observation of inland waterways and coastal areas.

The high-resolution multispectral imagery captured by the Sentinel-2 mission is useful for a broad range of applications, which includes monitoring of vegetation, soil, and water cover, as well as humanitarian and disaster risk.

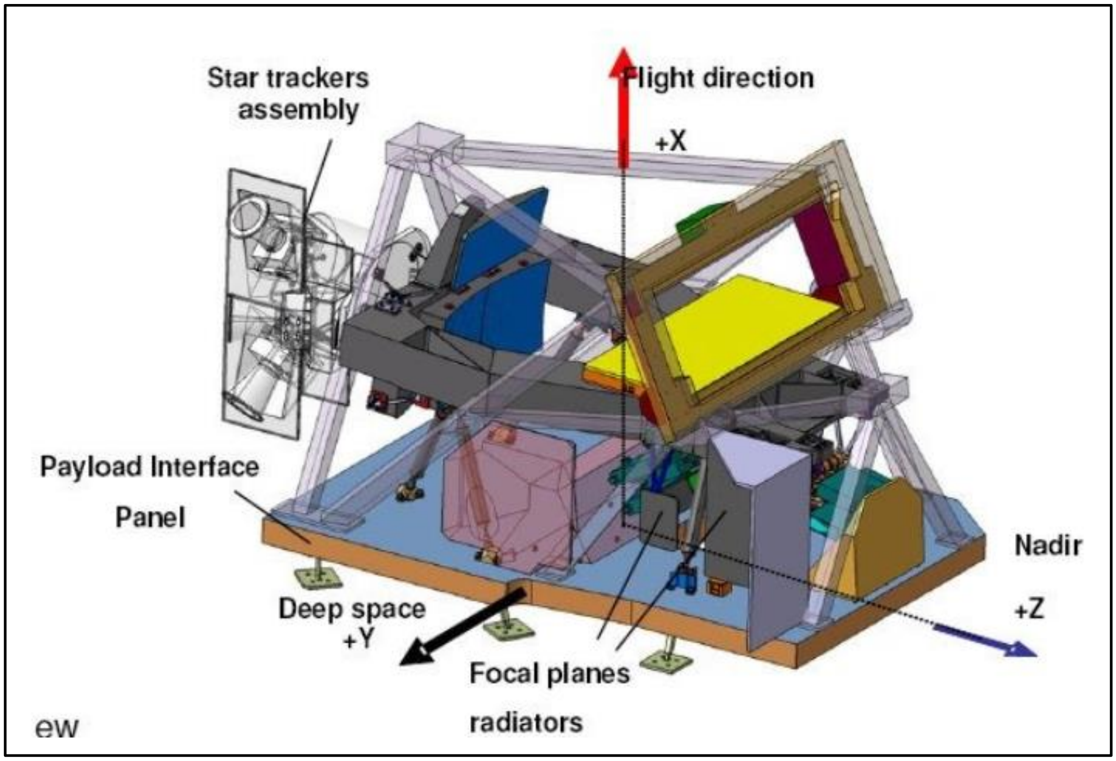
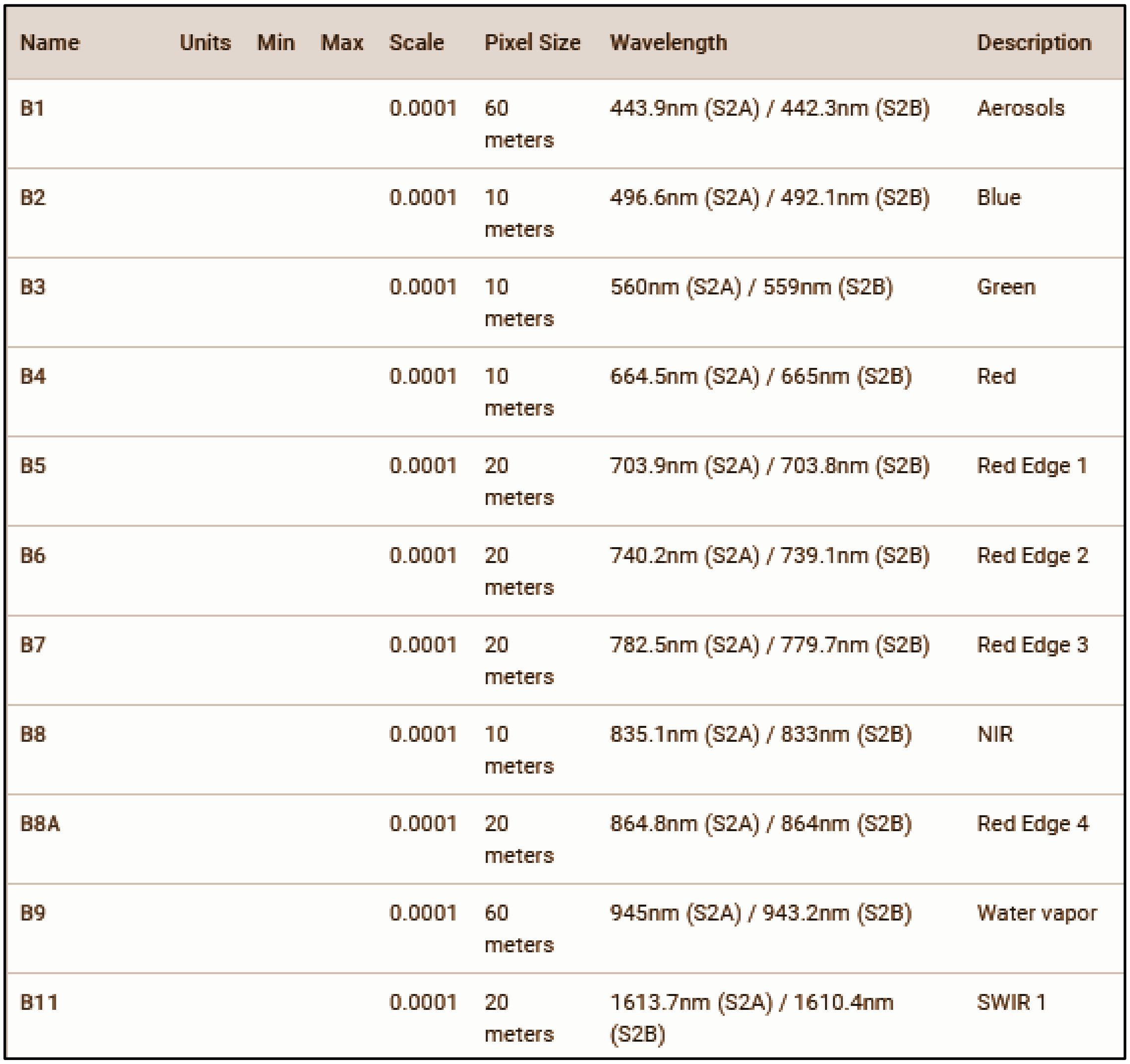


Fig 2.1 MSI instrument

2.2 Surface Reflectance

Surface Reflectance is the reflectance of the surface of the Earth. Clouds and other atmospheric components do not affect surface reflectance spectra. Surface reflectance improves comparison between multiple images over the same region by accounting for atmospheric effects such as aerosol scattering and thin clouds, which can help in the detection and characterization of Earth's surface change.

Dataset availability: 2017-03-28 – Present



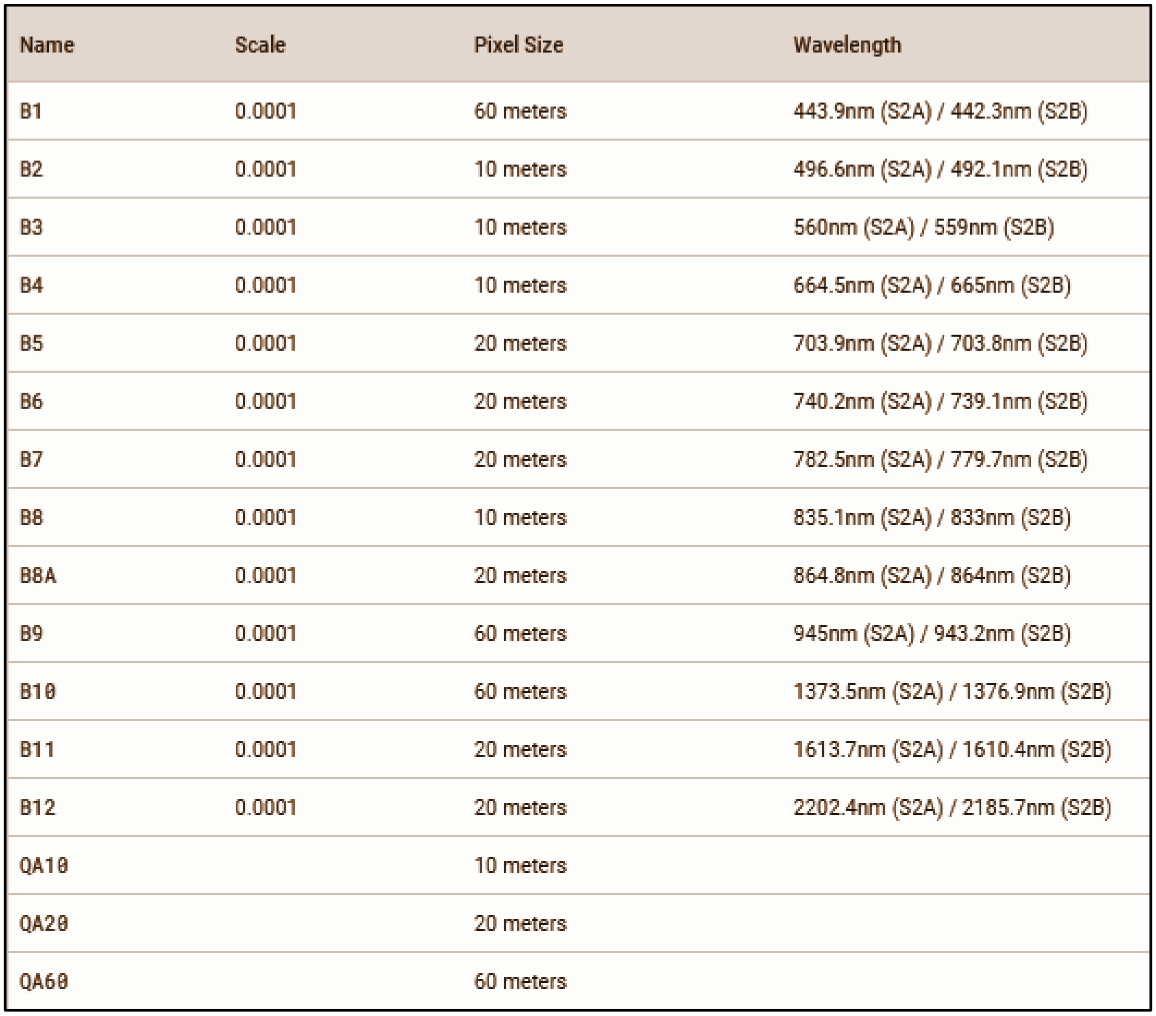
Name	Units	Min	Max	Scale	Pixel Size	Wavelength	Description
B1				0.0001	60 meters	443.9nm (S2A) / 442.3nm (S2B)	Aerosols
B2				0.0001	10 meters	496.6nm (S2A) / 492.1nm (S2B)	Blue
B3				0.0001	10 meters	560nm (S2A) / 559nm (S2B)	Green
B4				0.0001	10 meters	664.5nm (S2A) / 665nm (S2B)	Red
B5				0.0001	20 meters	703.9nm (S2A) / 703.8nm (S2B)	Red Edge 1
B6				0.0001	20 meters	740.2nm (S2A) / 739.1nm (S2B)	Red Edge 2
B7				0.0001	20 meters	782.5nm (S2A) / 779.7nm (S2B)	Red Edge 3
B8				0.0001	10 meters	835.1nm (S2A) / 833nm (S2B)	NIR
B8A				0.0001	20 meters	864.8nm (S2A) / 864nm (S2B)	Red Edge 4
B9				0.0001	60 meters	945nm (S2A) / 943.2nm (S2B)	Water vapor
B11				0.0001	20 meters	1613.7nm (S2A) / 1610.4nm (S2B)	SWIR 1

Fig 2.2 Sentinel-2 Surface Reflection Band Properties

2.3 Top of Atmosphere Reflectance

Top-of-atmosphere reflectance (or TOA reflectance) is the reflectance measured by a space-based sensor flying higher than the earth's atmosphere. These reflectance values will include contributions from clouds and atmospheric aerosols and gases.

Dataset availability: 2015-06-23 – Present



Name	Scale	Pixel Size	Wavelength
B1	0.0001	60 meters	443.9nm (S2A) / 442.3nm (S2B)
B2	0.0001	10 meters	496.6nm (S2A) / 492.1nm (S2B)
B3	0.0001	10 meters	560nm (S2A) / 559nm (S2B)
B4	0.0001	10 meters	664.5nm (S2A) / 665nm (S2B)
B5	0.0001	20 meters	703.9nm (S2A) / 703.8nm (S2B)
B6	0.0001	20 meters	740.2nm (S2A) / 739.1nm (S2B)
B7	0.0001	20 meters	782.5nm (S2A) / 779.7nm (S2B)
B8	0.0001	10 meters	835.1nm (S2A) / 833nm (S2B)
B8A	0.0001	20 meters	864.8nm (S2A) / 864nm (S2B)
B9	0.0001	60 meters	945nm (S2A) / 943.2nm (S2B)
B10	0.0001	60 meters	1373.5nm (S2A) / 1376.9nm (S2B)
B11	0.0001	20 meters	1613.7nm (S2A) / 1610.4nm (S2B)
B12	0.0001	20 meters	2202.4nm (S2A) / 2185.7nm (S2B)
QA10		10 meters	
QA20		20 meters	
QA60		60 meters	

Fig 2.3 Sentinel-2 Top-of-Atmosphere Reflectance Band Properties

Chapter 3

Algorithms and its Working

3.1 Random Forest

Use of Random Forests is an ensemble learning method. It can be used for classification and Regression and other tasks that operate by constructing a multitude of decision trees at training time. The final prediction for the classification problem is one that is pickled by the most number of trees within the forest. The mean or average prediction of the individual trees is returned if it is a regression task.

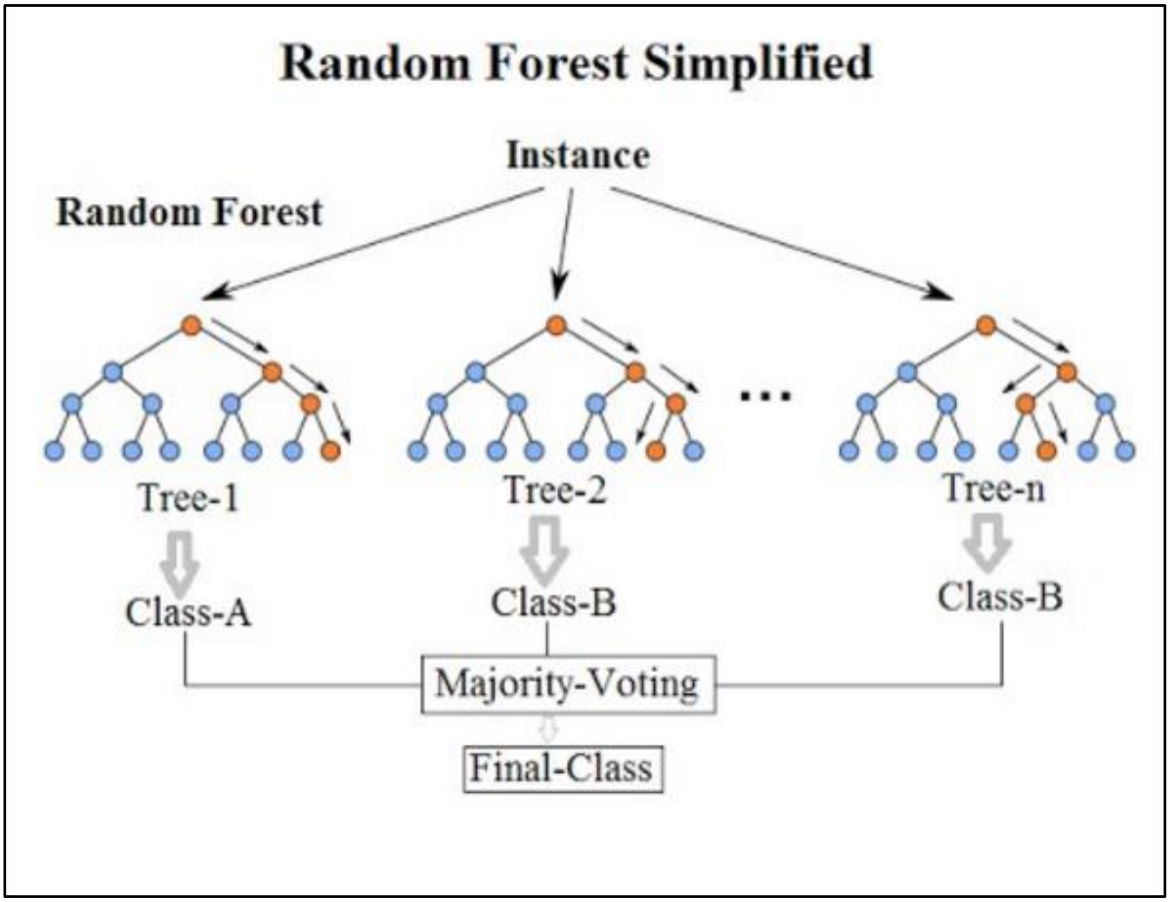


Fig 3.1 Working of random forest algorithm

3.2 Classification and Regression Trees(CART)

A CART model is represented in the form of a Binary tree. Each root node represents a single input parameter and its split point. The tree's leaf nodes include an output variable (y) that is utilized to produce a prediction. The tree can be saved as a graph or a series of steps in a file. Given a new input, the tree is traversed by evaluating the specific input that began at the tree's root node.

A learned binary tree is nothing more than a division of the input space. Consider each input variable to be a dimension on a p-dimensional space.

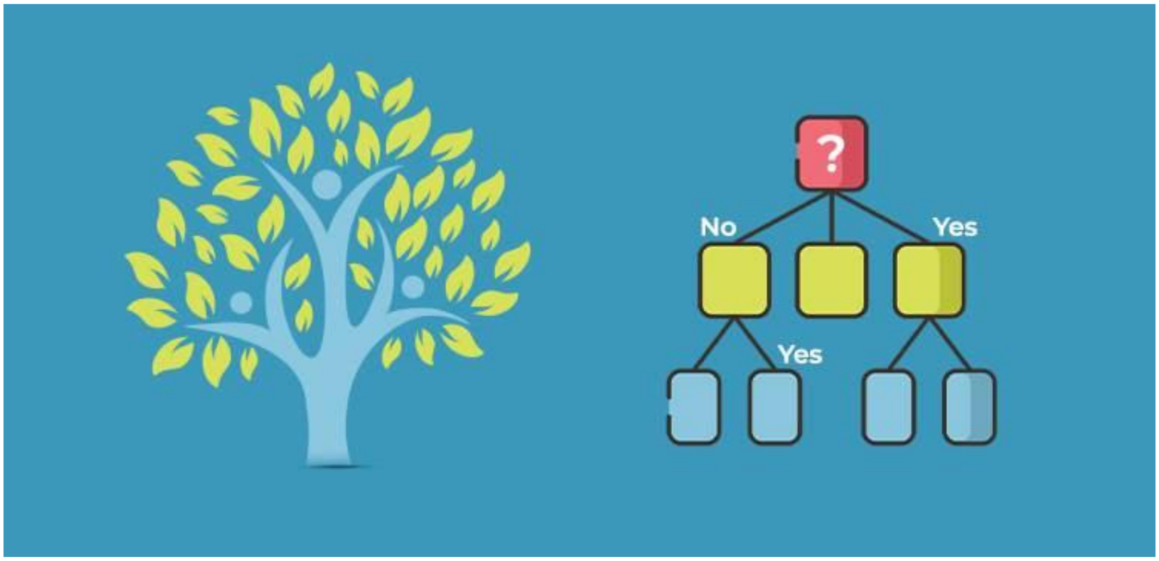


Fig 3.2 Working of classification and regression tree algorithm

3.3 Gradient Tree Boost

The gradient boosting approach may be used to forecast both continuous(as a regressor) and categorical target variables. If used as a regressor, the cost function is Mean Square Error (MSE), but when used as a classifier, the cost function is Log loss.

The cost function in regression problems is MSE, whereas the cost function in classification tasks is Log-Loss.

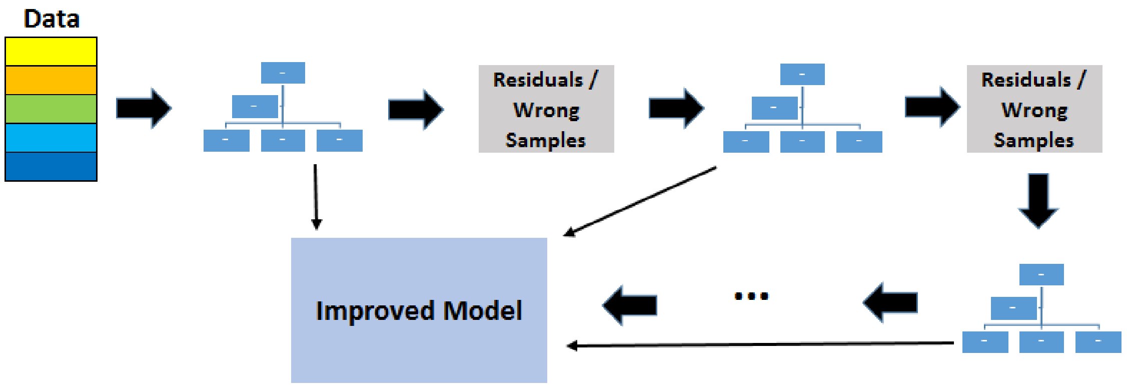


Fig 3.3 working of gradient tree boost algorithm

Chapter 4

Implementation

4.1 Data Collection

We were provided with reflectance spectra of in-situ data collected in Antarctica with the help of an ASD spectrometer.

The files included details for various parameters like

- Instrument
- Detectors
- Date: Time Temperature
- Battery Voltage
- Foreoptic
- Radiometric Calibration
- Wavelength Range
- Latitude
- Longitude
- Altitude
- GPS Time

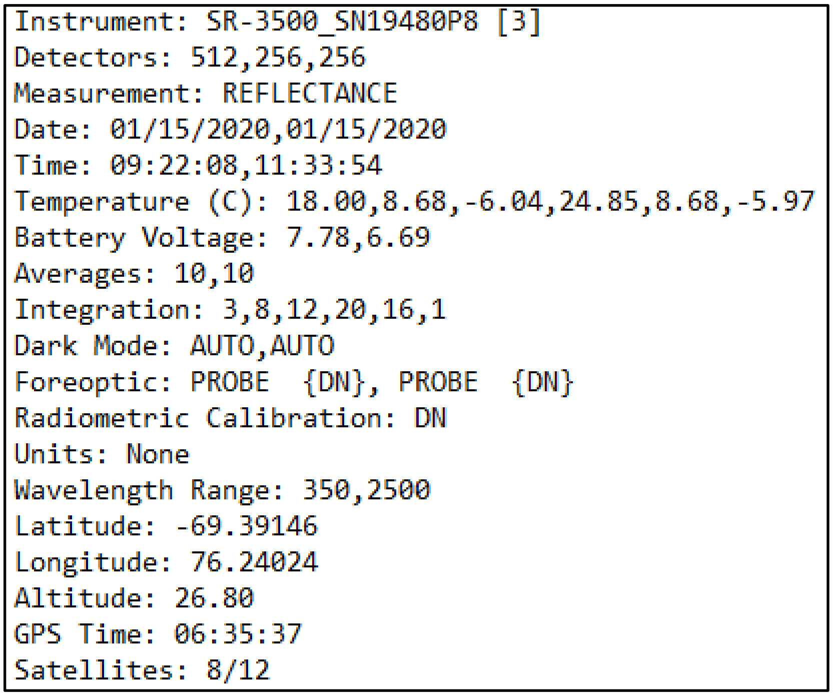
These samples were collected from

- Fischer North
- Bollingen
- Grovnes
- Manning Island
- McLeod Island
- Osmar Strait

The files included spectral signatures of

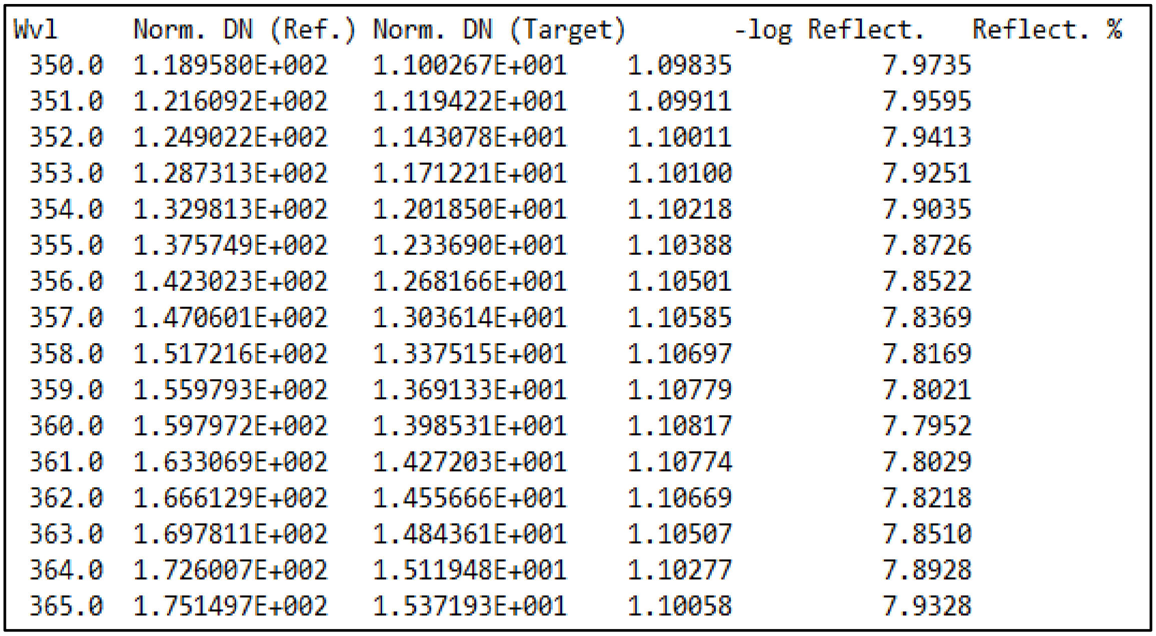
- Umbilicaria
- Moss
- Foliose

10 such files were provided to us initially. Picking up the coordinates from these files, we started exploring the Console and inspector tool in the Google Earth Engine.



```
Instrument: SR-3500_SN19480P8 [3]
Detectors: 512,256,256
Measurement: REFLECTANCE
Date: 01/15/2020,01/15/2020
Time: 09:22:08,11:33:54
Temperature (C): 18.00,8.68,-6.04,24.85,8.68,-5.97
Battery Voltage: 7.78,6.69
Averages: 10,10
Integration: 3,8,12,20,16,1
Dark Mode: AUTO,AUTO
Foreoptic: PROBE  {DN}, PROBE  {DN}
Radiometric Calibration: DN
Units: None
Wavelength Range: 350,2500
Latitude: -69.39146
Longitude: 76.24024
Altitude: 26.80
GPS Time: 06:35:37
Satellites: 8/12
```

These Files contain reflectance values(in %) for wavelengths starting from 350 nm to 2500 nm.

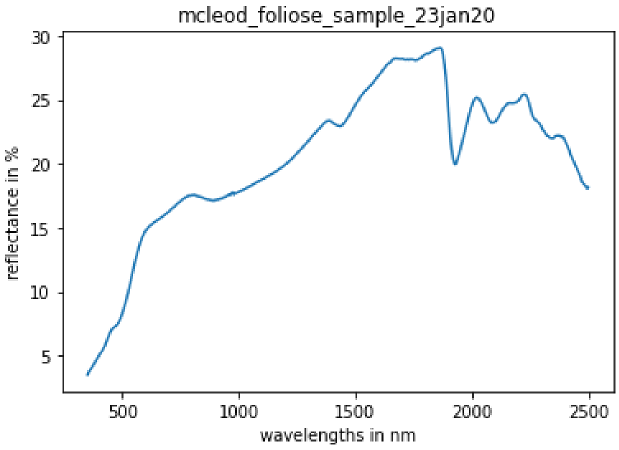


```
Wvl      Norm. DN (Ref.) Norm. DN (Target)       -log Reflect.   Reflect. %
 350.0  1.189580E+002   1.100267E+001    1.09835          7.9735
 351.0  1.216092E+002   1.119422E+001    1.09911          7.9595
 352.0  1.249022E+002   1.143078E+001    1.10011          7.9413
 353.0  1.287313E+002   1.171221E+001    1.10100          7.9251
 354.0  1.329813E+002   1.201850E+001    1.10218          7.9035
 355.0  1.375749E+002   1.233690E+001    1.10388          7.8726
 356.0  1.423023E+002   1.268166E+001    1.10501          7.8522
 357.0  1.470601E+002   1.303614E+001    1.10585          7.8369
 358.0  1.517216E+002   1.337515E+001    1.10697          7.8169
 359.0  1.559793E+002   1.369133E+001    1.10779          7.8021
 360.0  1.597972E+002   1.398531E+001    1.10817          7.7952
 361.0  1.633069E+002   1.427203E+001    1.10774          7.8029
 362.0  1.666129E+002   1.455666E+001    1.10669          7.8218
 363.0  1.697811E+002   1.484361E+001    1.10507          7.8510
 364.0  1.726007E+002   1.511948E+001    1.10277          7.8928
 365.0  1.751497E+002   1.537193E+001    1.10058          7.9328
```

4.2 Data Cleaning

First Step was to use mat-plot-lib in spyder IDE to plot Wavelength-Reflectance Line Graph for the points given in the initial configuration files, in an attempt to understand the spectral behavior of Lichen specimens. By doing so we were able to determine which bands would play an important role while segregating pixels based on their reflectance values available in the form of bands in the sentinel-2 dataset.

While choosing points to train an initial model we eliminated points that had different spectral behavior as compared to other points.

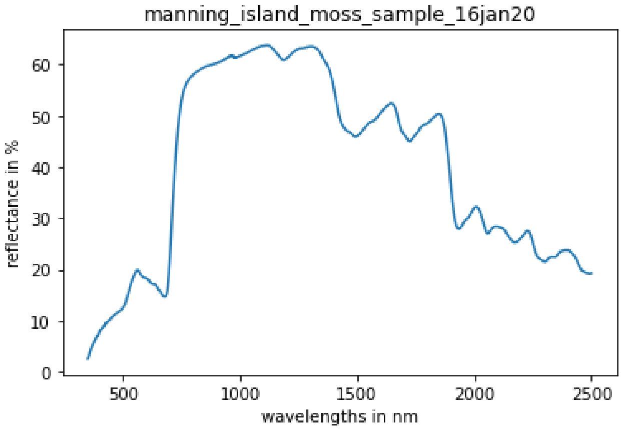


Instrument:SR-3500_SN19480P8
Measurement: Reflectance

Wavelength: 350nm - 2500 nm
Latitude: -69.37288
Longitude: 76.14082

X-axis: wavelength in nm.
Y-axis: reflectance in %

Fig 4.1 McLeod foliose sample

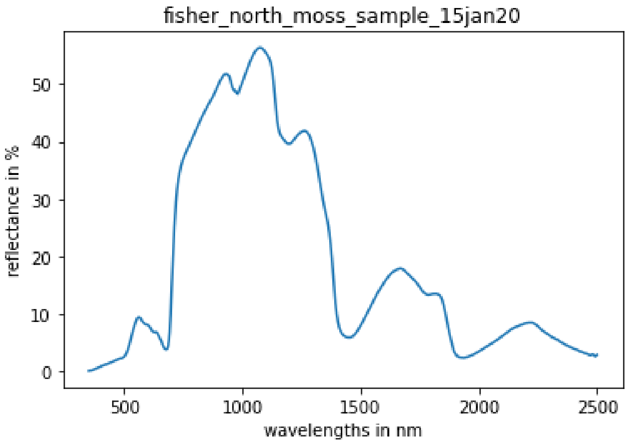


Instrument:SR-3500_SN19480P8
Measurement: Reflectance

Wavelength: 350nm - 2500 nm
Latitude: -69.35141
Longitude: 76.34597

X-axis: wavelength in nm.
Y-axis: reflectance in %

Fig 4.2 Manning island moss sample



Instrument:SR-3500_SN19480P8
Measurement: Reflectance

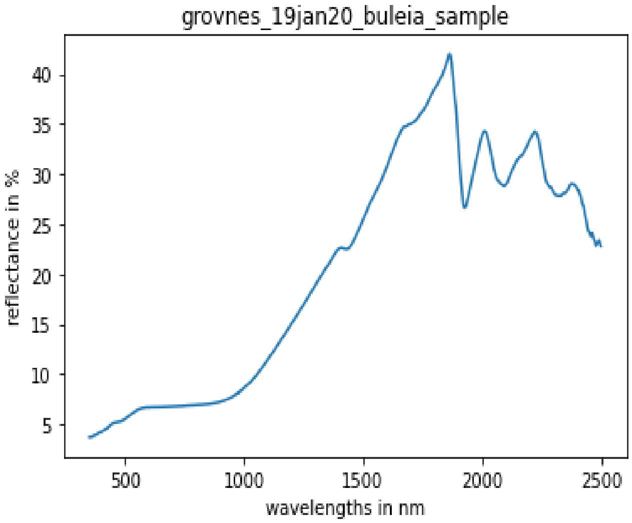
Wavelength: 350nm - 2500 nm
Latitude: -69.39066
Longitude: 76.24129

X-axis: wavelength in nm.
Y-axis: reflectance in %

Fig 4.3 Fisher north moss sample

4.3 Data Analysis and Visualization

Following are the graphs where we observed a pattern of peaks and drops. We observed that the reflectance value peaks at around 1850 nm before a sudden dip which forms a bottom at about 2000nm. A peak at a certain wavelength indicates that on being exposed to radiation of that wavelength, a majority of waves are reflected back and recaptured. A value of 10% reflectance, for instance, at 1000nm on the x-axis indicates that 90% of the total incident waves were absorbed and 10% were reflected back.

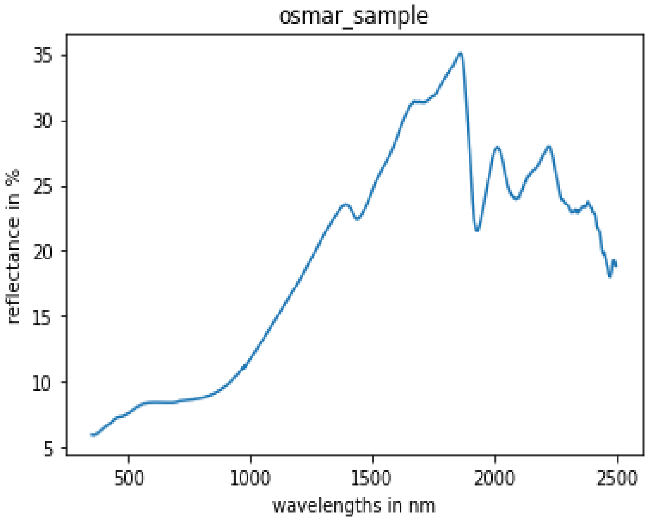


Instrument:SR-3500_SN19480P8
Measurement: Reflectance

Wavelength: 350nm - 2500 nm
Latitude: -69.40349
Longitude: 76.19136

X-axis: wavelength in nm.
Y-axis: reflectance in %

Fig 4.4 Grovens Bulaia sample

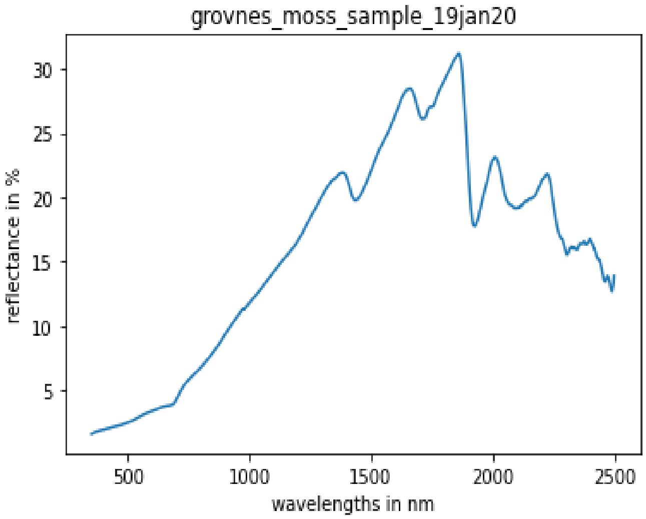


Instrument:SR-3500_SN19480P8
Measurement: Reflectance

Wavelength: 350nm - 2500 nm
Latitude: -69.37386
Longitude: 76.01898

X-axis: wavelength in nm.
Y-axis: reflectance in %

Fig 4.5 Osmar sample

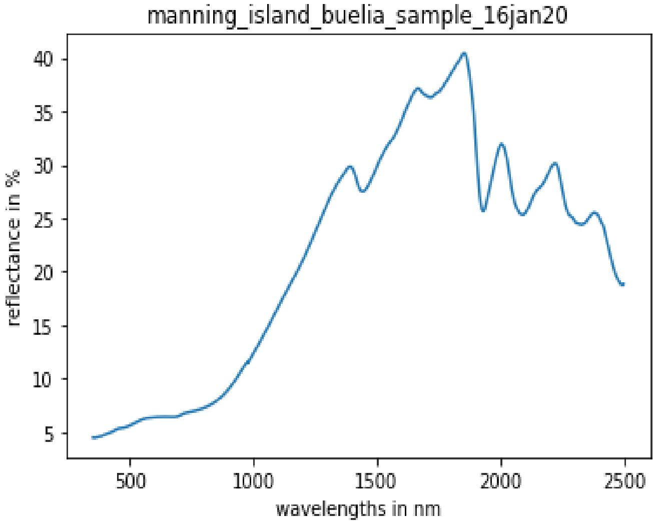


Instrument:SR-3500_SN19480P8
Measurement: Reflectance

Wavelength: 350nm - 2500 nm
Latitude: -69.40391
Longitude: 76.19292

X-axis: wavelength in nm.
Y-axis: reflectance in %

Fig 4.6 Groven's moss sample

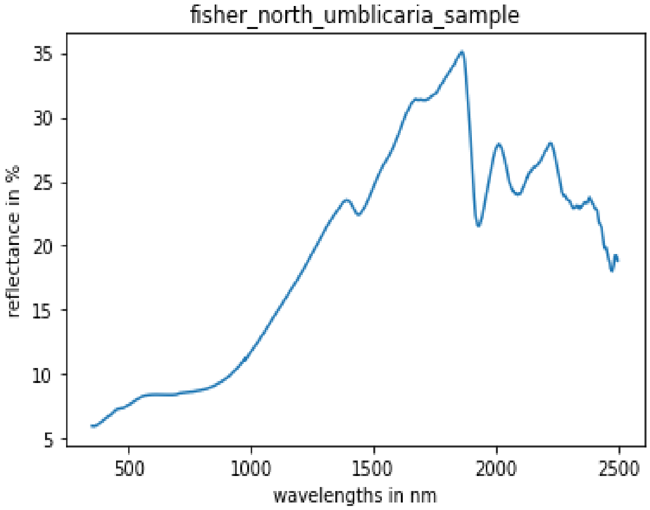


Instrument:SR-3500_SN19480P8
Measurement: Reflectance

Wavelength: 350nm - 2500 nm
Latitude: -69.35683
Longitude: 76.32302

X-axis: wavelength in nm.
Y-axis: reflectance in %

Fig 4.7 Manning island Buelia sample



Instrument:SR-3500_SN19480P8
Measurement: Reflectance

Wavelength: 350nm - 2500 nm
Latitude: -69.39072
Longitude: 76.24108

X-axis: wavelength in nm.
Y-axis: reflectance in %

Fig 4.8 Fisher north umblicaria sample

4.4 Training (North Deception Island)

As discussed earlier, our main study area is Larsman Hills. But before moving on to that, we intended to compare algorithms we could possibly use. In order to do this, we focussed on finding out a region that was already well-documented. With the help of one of the research papers that our external guide provided us with, we were able to find out a region that was mapped for the presence of lichen.

The name of the area is Deception Island. Deception Island can be mainly divided into 2 main parts: North Deception Island and South Deception Island. So we first started looking for cloud/cirrus-free sentinel-2 images for this area. Lesser the cover of snow, the more the study area. With this in mind, we found the image collection for our area of interest and filtered it by Antarctic summer i.e the date range of October 2020 to February 2021.

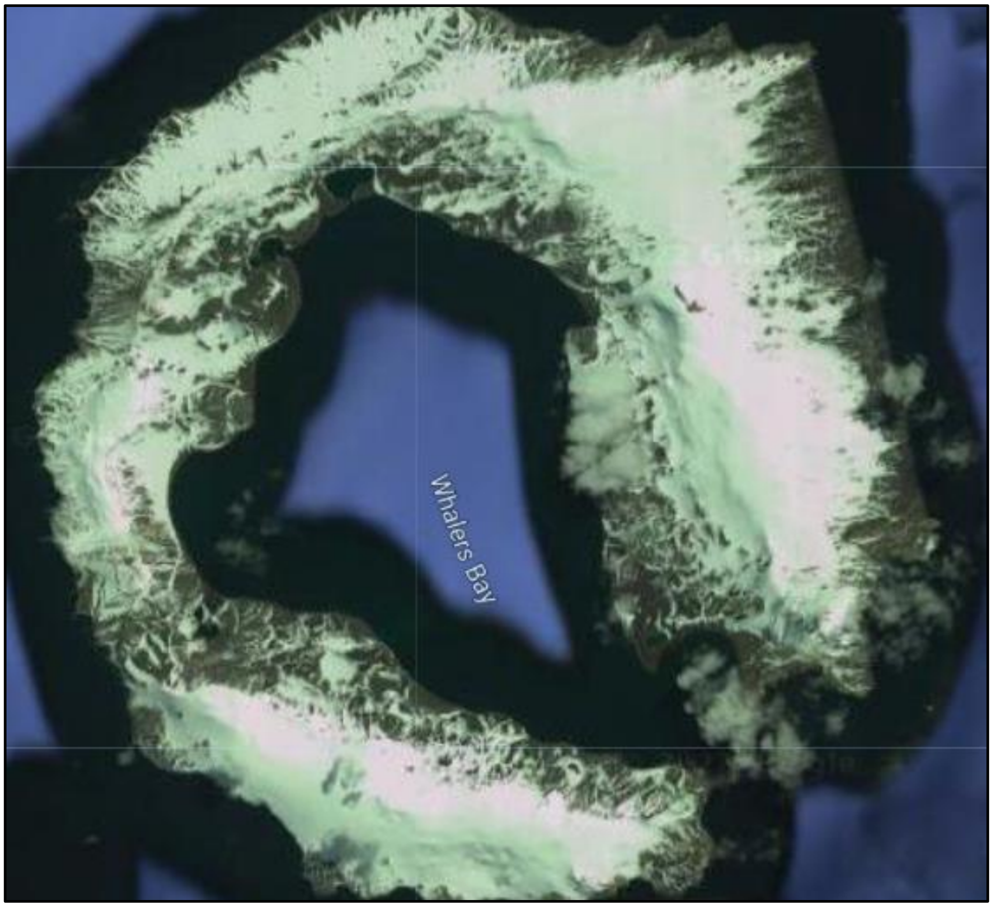
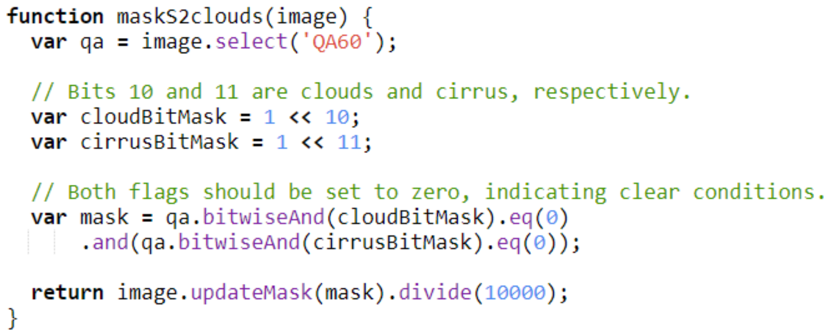


Fig 4.9 Deception Island (Satellite image)

After getting the desired image collection, We used a median reducer to generate a single image from the image collection on which we would use machine learning algorithms. In order to account for the cloudy pixels from that image, we used cloud masking with the help of the 'QA60' band of the sentinel-2 product.



```
function maskS2clouds(image) {
  var qa = image.select('QA60');

  // Bits 10 and 11 are clouds and cirrus, respectively.
  var cloudBitMask = 1 << 10;
  var cirrusBitMask = 1 << 11;

  // Both flags should be set to zero, indicating clear conditions.
  var mask = qa.bitwiseAnd(cloudBitMask).eq(0)
      .and(qa.bitwiseAnd(cirrusBitMask).eq(0));

  return image.updateMask(mask).divide(10000);
}
```

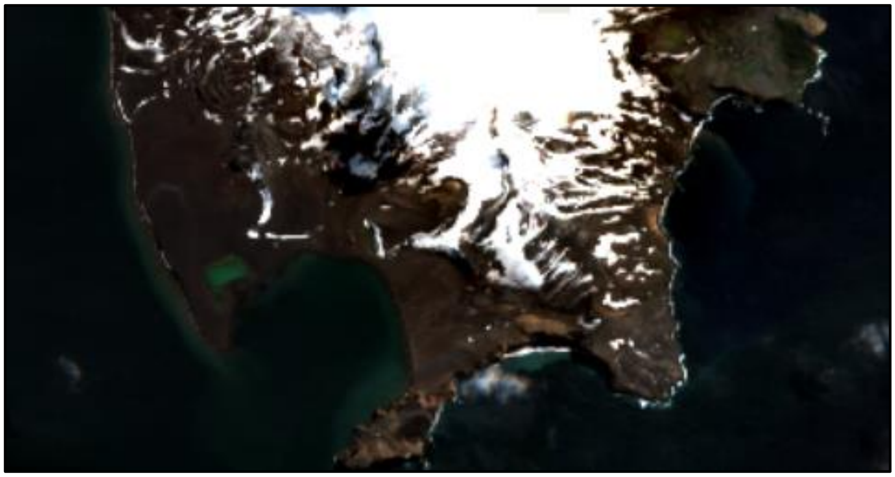


Fig 4.10 Northside of Deception Island

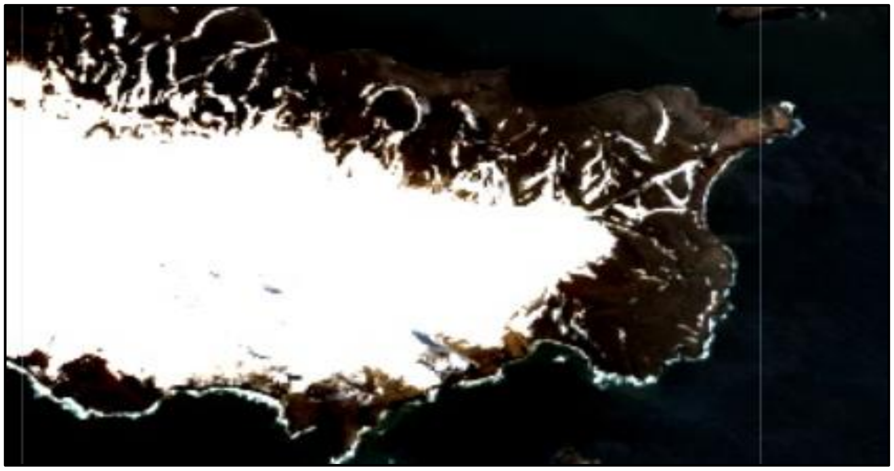
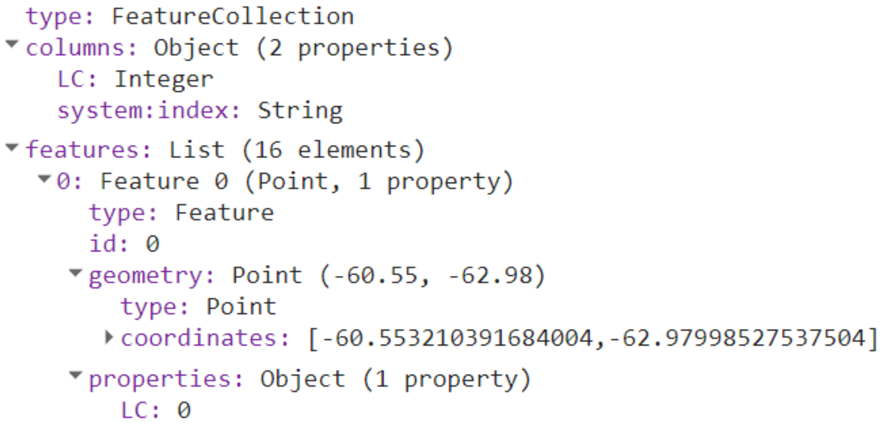


Fig 4.11 Southside of Deception Island

In order to compare the algorithms we first collected the training data points on north deception island and made feature collections. The data points can be divided into four classes snow, water, lichen, and not lichen.

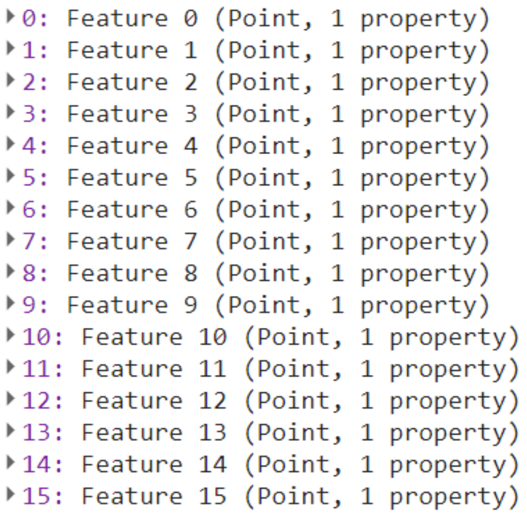
There are three attributes in the feature collection. The first one is “type” which indicates that this is ‘FeatureCollection’. The second one is “columns”. In our case there are two columns, one is class ID number (Integer) and the second is system: index. The third attribute is “features” which contains geometry points, coordinates and properties.



```
  type: FeatureCollection
▾ columns: Object (2 properties)
    LC: Integer
    system:index: String
▾ features: List (16 elements)
  ▾ 0: Feature 0 (Point, 1 property)
      type: Feature
      id: 0
    ▾ geometry: Point (-60.55, -62.98)
        type: Point
      ▸ coordinates: [-60.553210391684004,-62.97998527537504]
    ▾ properties: Object (1 property)
        LC: 0
```

Fig 4.12 Insights of feature collection

We collected 16 points of lichen pixels, 35 points of non-lichens pixels, 26 points of water pixels, and 13 points of snow pixels.



```
▸ 0: Feature 0 (Point, 1 property)
▸ 1: Feature 1 (Point, 1 property)
▸ 2: Feature 2 (Point, 1 property)
▸ 3: Feature 3 (Point, 1 property)
▸ 4: Feature 4 (Point, 1 property)
▸ 5: Feature 5 (Point, 1 property)
▸ 6: Feature 6 (Point, 1 property)
▸ 7: Feature 7 (Point, 1 property)
▸ 8: Feature 8 (Point, 1 property)
▸ 9: Feature 9 (Point, 1 property)
▸ 10: Feature 10 (Point, 1 property)
▸ 11: Feature 11 (Point, 1 property)
▸ 12: Feature 12 (Point, 1 property)
▸ 13: Feature 13 (Point, 1 property)
▸ 14: Feature 14 (Point, 1 property)
▸ 15: Feature 15 (Point, 1 property)
```

We have a total of 16 points for lichen (shown on the left), 35 points for the not-lichen class, 26 for water, and 13 for snow.

Total points: 16+25+36+14 = 90

Fig 4.13 16 points of lichens.

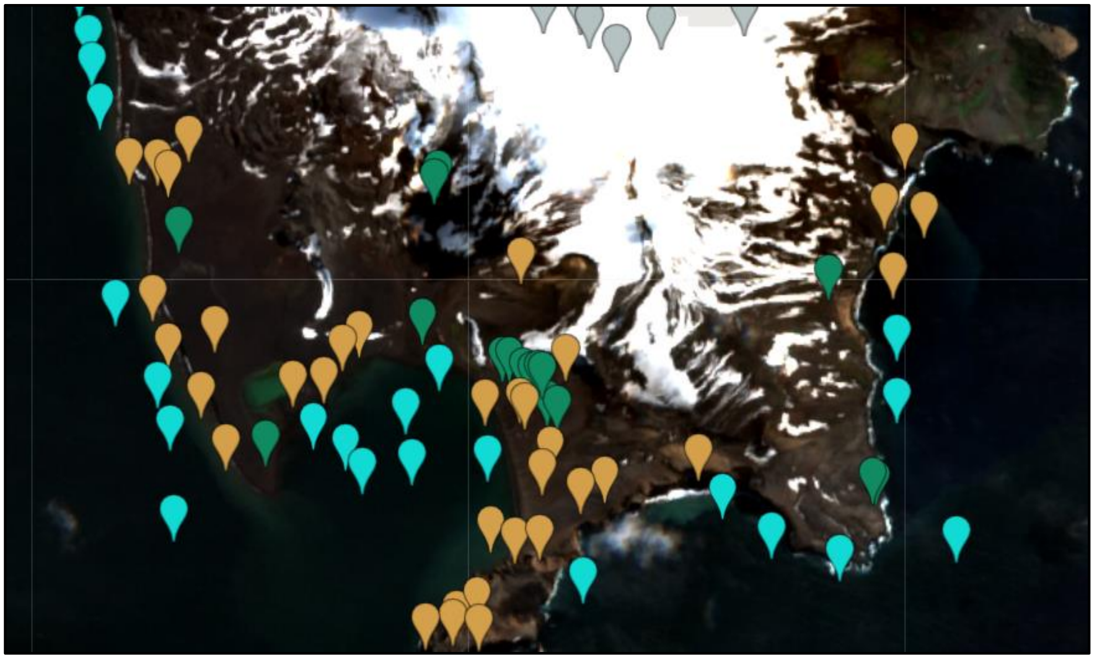


Fig 4.14 Mapped training points on the north side of the island.

As we discussed earlier in Chapter 2 of the 2.2 Surface Reflections part, we have 12 bands in total (B2, B3, B4, B5, B6, B7, B8, B8A, B9, B11, B12, QA60). In order to efficiently segregate between pixels that are snow and those which are not, we added one more band which is known as NDSI (normalized difference snow index).

When compared to Fractional Snow Cover (FSC), the Normalized Difference Snow Index (NDSI) snow cover is an index that provides a more accurate representation of snow detection. Snow has a very high visible (VIS) reflectance and a very low shortwave infrared (SWIR) reflectance, which is used to identify snow by separating it from most cloud types.

In order to calculate the NDSI ratio in our sentinel-2 product, we need two bands B3 and B11. (The range of the ratio is from 0 to 1. generally, if the ratio is more than 0.42 then those pixels can be considered as snow)

$$\text{Normalised Difference Snow Index } (NDSI) = \frac{Band\,3 - Band\,11}{Band\,3 + Band\,11}$$

Fig 4.15 NDSI ratio

After calculating the NDSI ratio we added it to our image as a band in order to train the model. As we discussed earlier, if the value of the NDSI ratio is more than 0.42 then we can safely conclude that the highest single majority cover for that pixel is snow. For our use case, we cannot simply discard all sich pixels as the aim is to find traces of lichen's spectral response in a pixel. We also can not divide the pixel further because the pixel is the smallest unit of the image. So, instead of putting a threshold, we added NDSI as a band so that it can be considered as an attribute during the training phase. Now we have 13 bands for training. Then we merged all four feature collections in order to train the model.

Firstly we sample the input imagery to get a feature collection of training data. Then we train different supervised machine learning models in order to compare outcomes.

4.4.1 CART (classification and Regression Tree)

First we trained using the CART (classification and regression tree) algorithm. In order to evaluate the outcomes, we checked the resubstitution accuracy and confusion matrix. In the google earth engine, the number of leaf nodes is set to 'very high' for the CART algorithm.

<table>
<tr><th>Method</th><th>Confusion matrix</th><th>Resub-Accuracy</th></tr>
<tr><td>Default parameter (No limit in leaf nodes) and we try 90, 80, up to 20 number of leaf nodes.</td><td>
<table>
<tr><th></th><th>lichen</th><th>not-lichen</th><th>water</th><th>snow</th></tr>
<tr><td>lichen</td><td>16</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>not-lichen</td><td>0</td><td>52</td><td>0</td><td>0</td></tr>
<tr><td>water</td><td>0</td><td>0</td><td>26</td><td>0</td></tr>
<tr><td>snow</td><td>0</td><td>0</td><td>0</td><td>13</td></tr>
</table>
</td><td>100 %</td></tr>
</table>

Fig 4.16 Resubstitution accuracy for CART algorithm (default)

The model is going through an overfitting problem, generally, overfitting happens when we have less number of data or we trained it through a very high number of calculations. To overcome it we can not reduce the data because every feature is important for us that’s why we reduce the number of calculations by reducing the number of leaf nodes.

<table>
<tr><th>Method</th><th>Confusion matrix</th><th>Resub-
Accuracy</th></tr>
<tr><td>Number of the leaf node is 10</td><td>
<table>
<tr><th></th><th>lichen</th><th>not-lichen</th><th>water</th><th>snow</th></tr>
<tr><td>lichen</td><td>10</td><td>6</td><td>0</td><td>0</td></tr>
<tr><td>not-lichen</td><td>1</td><td>50</td><td>0</td><td>1</td></tr>
<tr><td>water</td><td>0</td><td>0</td><td>26</td><td>0</td></tr>
<tr><td>snow</td><td>0</td><td>0</td><td>0</td><td>13</td></tr>
</table>
</td><td>92.52 %</td></tr>
</table>

Fig 4.17 Resubstitution accuracy for CART algorithm (10 leaf nodes)

4.4.2 RF (Random Forest)

After the CART algorithm in order to get the better result we trained the random forest model with different values as a number of tree parameter. We checked for 10, 30, 50 trees. We also checked with more number of trees but the results came similar to the result of 50 trees.

<table>
<tr><th>Method</th><th>Confusion matrix</th><th>Resub-Accuracy</th></tr>
<tr><td>Number of trees are 10</td><td><table>
<tr><th></th><th>lichen</th><th>not-lichen</th><th>water</th><th>snow</th></tr>
<tr><td>lichen</td><td>14</td><td>2</td><td>0</td><td>0</td></tr>
<tr><td>not-lichen</td><td>2</td><td>50</td><td>0</td><td>0</td></tr>
<tr><td>water</td><td>0</td><td>1</td><td>25</td><td>0</td></tr>
<tr><td>snow</td><td>0</td><td>0</td><td>0</td><td>13</td></tr>
</table></td><td>95.32 %</td></tr>
</table>

Fig 4.18 Resubstitution accuracy for RF algorithm (10 trees)

<table>
<tr><th>Method</th><th>Confusion matrix</th><th>Resub-Accuracy</th></tr>
<tr><td>Number of trees are 30</td><td><table>
<tr><th></th><th>lichen</th><th>not-lichen</th><th>water</th><th>snow</th></tr>
<tr><td>lichen</td><td>15</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>not-lichen</td><td>0</td><td>51</td><td>0</td><td>1</td></tr>
<tr><td>water</td><td>0</td><td>0</td><td>26</td><td>0</td></tr>
<tr><td>snow</td><td>0</td><td>0</td><td>0</td><td>13</td></tr>
</table></td><td>97.12 %</td></tr>
</table>

Fig 4.19 Resubstitution accuracy for RF algorithm (30 trees)

<table>
<tr><th>Method</th><th>Confusion matrix</th><th>Resub-
Accuracy</th></tr>
<tr><td>Number of trees are 50</td><td>
<table>
<tr><th></th><th>lichen</th><th>not-lichen</th><th>water</th><th>snow</th></tr>
<tr><td>lichen</td><td>15</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>not-lichen</td><td>0</td><td>51</td><td>0</td><td>1</td></tr>
<tr><td>water</td><td>0</td><td>0</td><td>26</td><td>0</td></tr>
<tr><td>snow</td><td>0</td><td>0</td><td>0</td><td>13</td></tr>
</table>
</td><td>98.13 %</td></tr>
</table>

Fig 4.20 Resubstitution accuracy for RF algorithm (50 trees)

4.5 Testing (South Deception Island)

In order to check which algorithm yields better results on unseen data, we tested our model on southern deception island. With the help of coordinates mentioned in “Mapping lichen distribution on the Antarctic Peninsula using remote sensing, lichen spectra and photographic documentation by citizen scientists’ published in Polar Research, we were able to find 4 points on which the model can be tested. For validation, we needed more points and the goal here was just to gain first-hand experience of using various validation techniques of Google Earth Engine. For this purpose, we decided to take a few sample points by manually matching the band value curves as seen in the console. We temporarily added such 13 pixels as lichen, 12 pixels as not lichen, 10 pixels as water, and 14 pixels as snow.

4.5.1 CART (classification and Regression Tree)

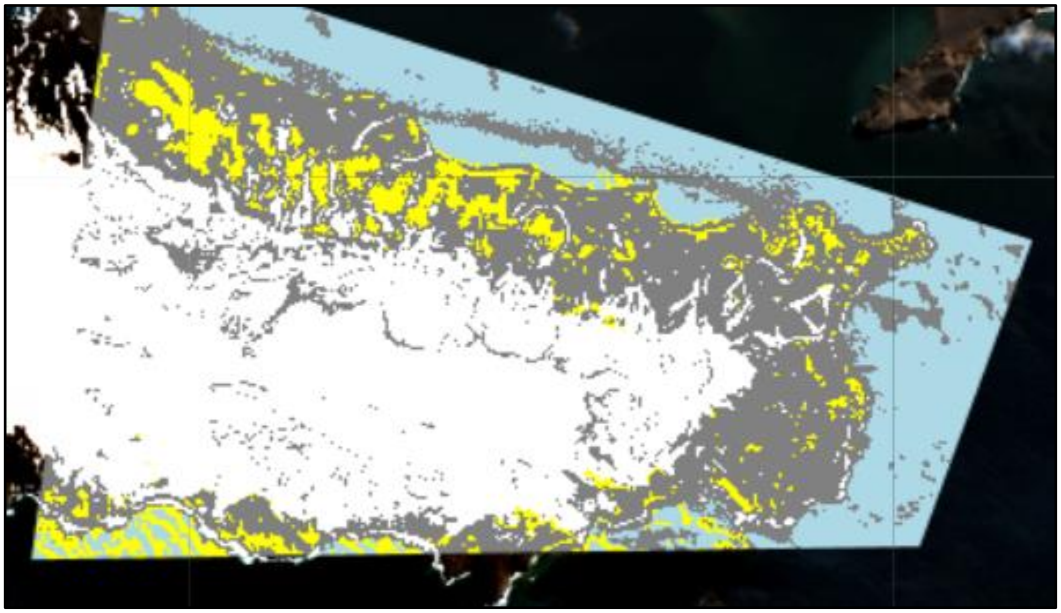


Fig 4.21 Classification of the southside of deception island (default)

Method	Confusion matrix					Test Accuracy
Default parameter (No limit in leaf nodes) and we try 90, 80, up to 20 number of leaf nodes.		lichen	not-lichen	water	snow	72.32 %
	lichen	3	9	0	1	
	not-lichen	1	8	0	1	
	water	0	0	10	0	
	snow	0	1	0	13	

Fig 4.22 Test accuracy for CART algorithm (default)

As we go for less number of leaf nodes such as 10 then we came up with less accuracy than that mentioned above.

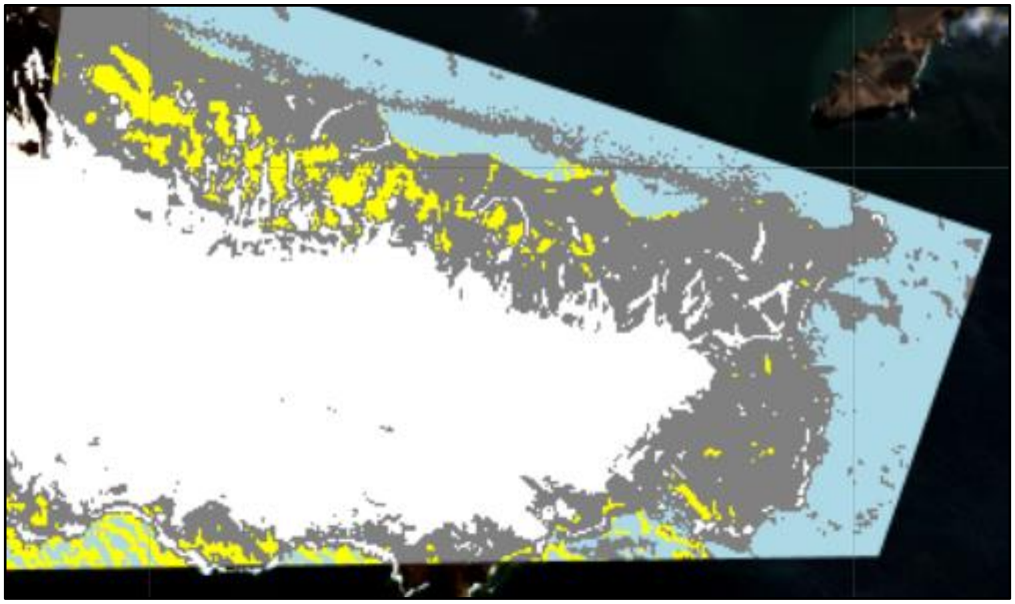


Fig 4.23 Classification of the southside of deception island (10 leaf nodes)

<table>
<tr><th>Method</th><th>Confusion matrix</th><th>Test Accuracy</th></tr>
<tr><td>Number of the leaf node is 10</td><td>
<table>
<tr><th></th><th>lichen</th><th>not-lichen</th><th>water</th><th>snow</th></tr>
<tr><td>lichen</td><td>1</td><td>11</td><td>0</td><td>1</td></tr>
<tr><td>not-lichen</td><td>1</td><td>10</td><td>0</td><td>1</td></tr>
<tr><td>water</td><td>0</td><td>0</td><td>10</td><td>0</td></tr>
<tr><td>snow</td><td>0</td><td>0</td><td>0</td><td>14</td></tr>
</table>
</td><td>71.42 %</td></tr>
</table>

Fig 4.24 Test accuracy for CART algorithm (10 leaf node)

4.5.1 RF (Random Forest)

Using similar data points, we repeated the process explained above, but this time we used a Random Forest Model. We observed that Radom Forest provided better results and generated higher accuracy outputs. so we decided to implement a Random Forest algorithm for our study area(Larsemann hills).

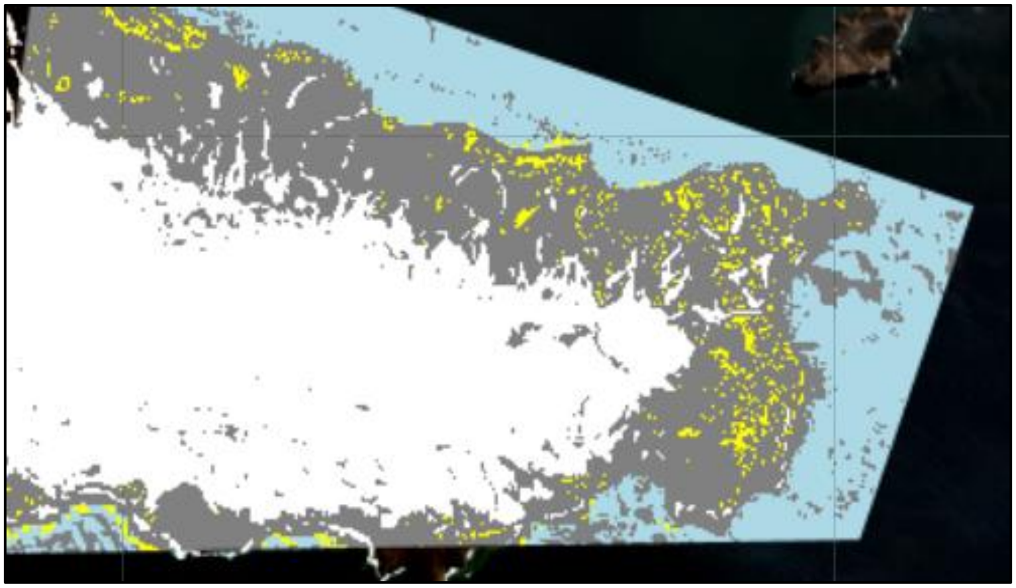


Fig 4.25 Classification of the southside of deception island (10 trees)

<table>
<tr><th>Method</th><th>Confusion matrix</th><th>Test Accuracy</th></tr>
<tr><td>Number of trees are 10</td><td>
<table>
<tr><th></th><th>lichen</th><th>not-lichen</th><th>water</th><th>snow</th></tr>
<tr><td>lichen</td><td>8</td><td>4</td><td>0</td><td>1</td></tr>
<tr><td>not-lichen</td><td>1</td><td>10</td><td>0</td><td>1</td></tr>
<tr><td>water</td><td>0</td><td>0</td><td>10</td><td>0</td></tr>
<tr><td>snow</td><td>0</td><td>0</td><td>0</td><td>14</td></tr>
</table>
</td><td>85.71 %</td></tr>
</table>

Fig 4.26 Test accuracy for RF algorithm (10 trees)

4.6 Training (South Antarctic) for Larsemann hills

After experimenting with various Algorithms, we finally started working on implementing Random Forest Model to produce output for Larsemann hills. We had around 1000 points from various sites in the Antarctic that include, but are not limited to Mcleod, fisher north, manning.

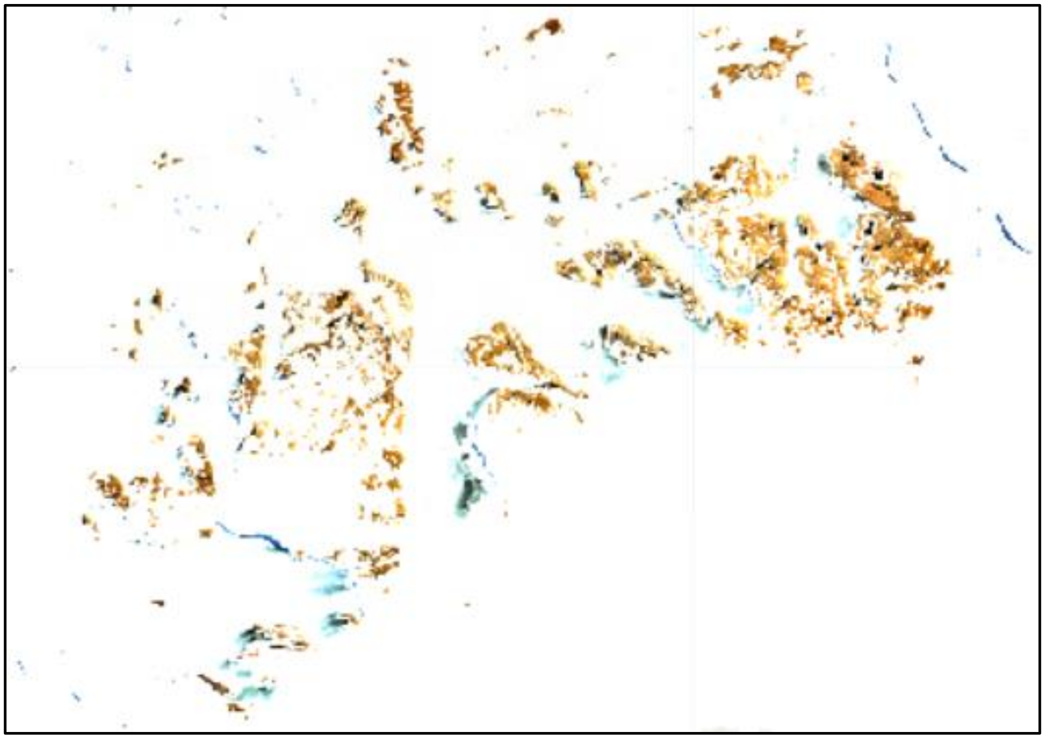


Fig 4.27 Larsemann hills (study area)

There is a possibility that a few points out of these thousand points that, at the time of collection were not covered by snow, are currently covered by a layer of snow. So the first step was to plot these points on the GEE image used for training and eliminate all the points which have an NDSI ratio of more than 0.42.

We collected 67 points from the south of larsemann hills and then other 25 points from larsemann hills itself. We decided to train the model on these points, and save some points from larsemann hills for validation.

We were also given 20 pixels to represent the class ‘bare rocks’ which are exposed rocks that had no traces of lichens.

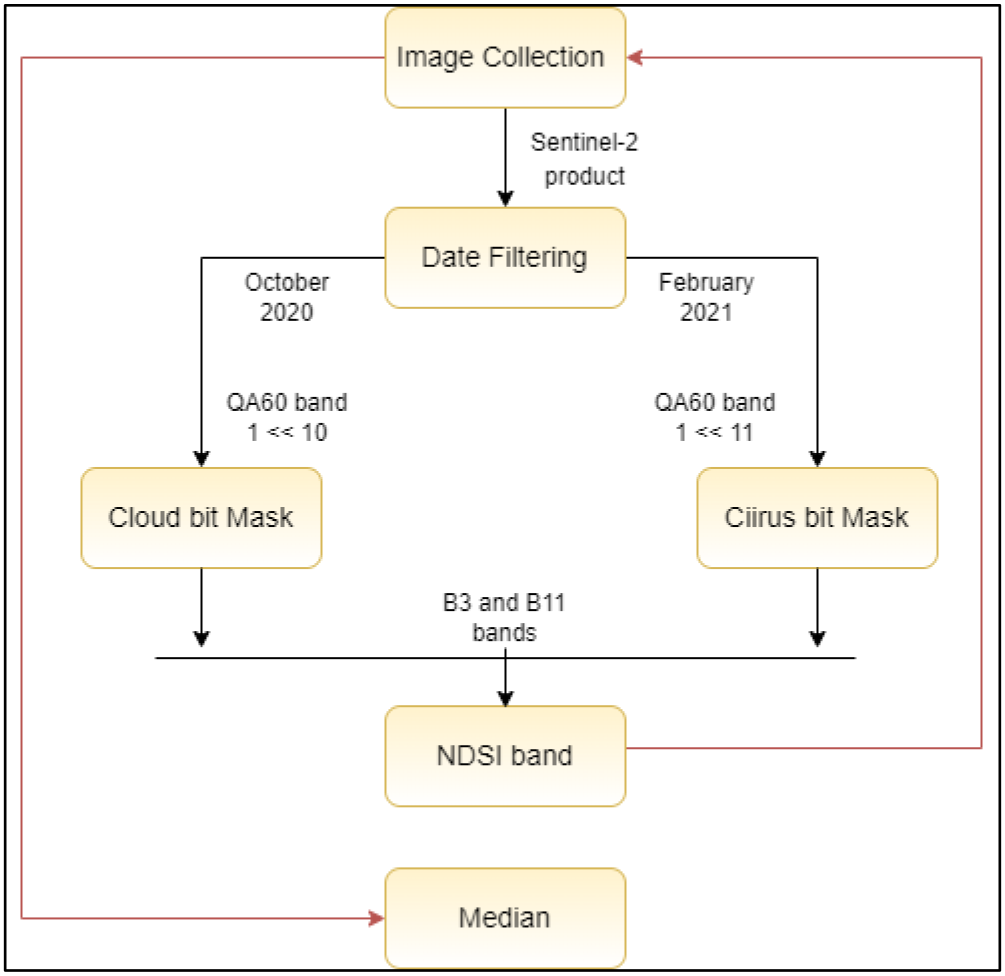


Fig 4.28 Single image workflow

The total number of training pixels:

Lichens	92
Bare_rocks	20
Snow	74

After generating a median image, we merged the training points and selected the said 13 bands. Then create sampled regions of interest and trained machine learning models.

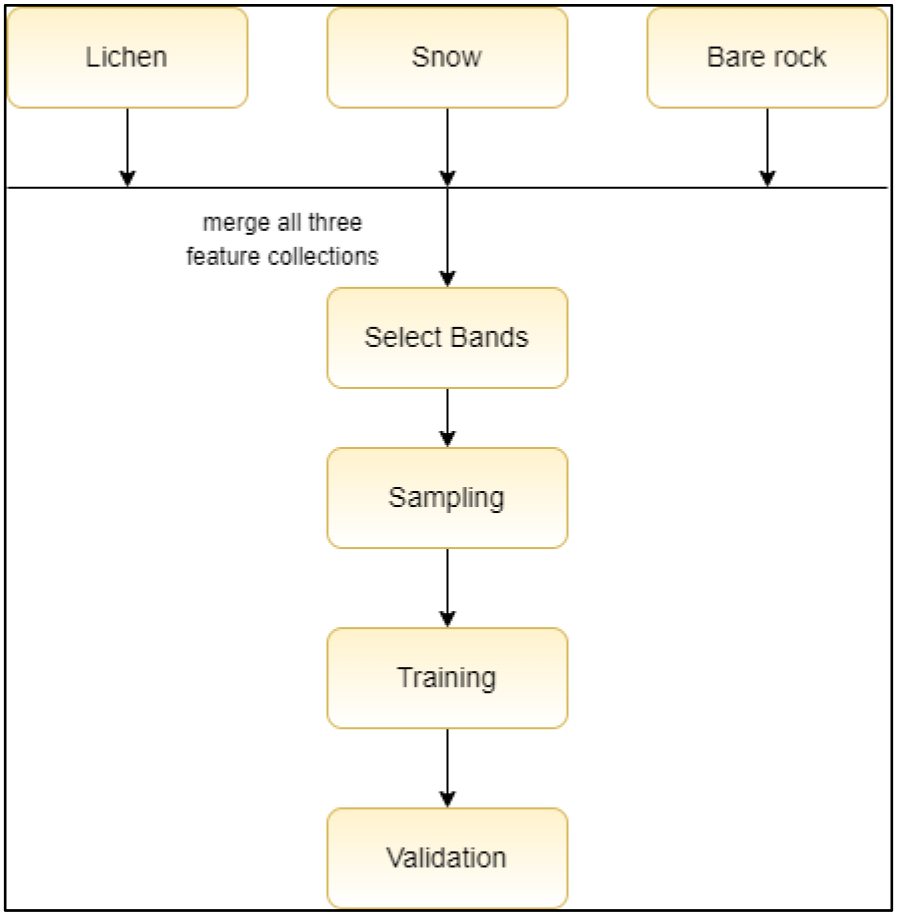


Fig 4.29 Model training workflow

Following are the outcomes for the model trained using Random Forest (along with a number of trees used).

4.6.1 Random forest with a number of trees are 10.

<table>
<tr><th>Method</th><th>Confusion matrix</th><th>Resub-Accuracy</th></tr>
<tr><td>Number of trees are 10</td><td>
<table>
<tr><th></th><th>lichen</th><th>bare_rocks</th><th>snow</th></tr>
<tr><td>lichen</td><td>90</td><td>0</td><td>2</td></tr>
<tr><td>bare_rocks</td><td>4</td><td>3</td><td>13</td></tr>
<tr><td>snow</td><td>0</td><td>74</td><td>0</td></tr>
</table>
</td><td>95.71 %</td></tr>
</table>

Fig 4.30 Resubstitution accuracy for RF algorithm (hills, 10 trees)

4.6.2 Random forest with a number of trees are 20.

<table>
<tr><th>Method</th><th>Confusion matrix</th><th>Resub-Accuracy</th></tr>
<tr><td>Number of trees are 20</td><td><table>
<tr><th></th><th>lichen</th><th>bare_rocks</th><th>snow</th></tr>
<tr><td>lichen</td><td>91</td><td>0</td><td>1</td></tr>
<tr><td>bare_rocks</td><td>1</td><td>2</td><td>17</td></tr>
<tr><td>snow</td><td>0</td><td>74</td><td>0</td></tr>
</table></td><td>97.81 %</td></tr>
</table>

Fig 4.31 Resubstitution accuracy for RF algorithm (hills, 20 trees)

4.6.3 Random forest with a number of trees are 30.

<table>
<tr><th>Method</th><th>Confusion matrix</th><th>Resub-Accuracy</th></tr>
<tr><td>Number of trees are 30</td><td><table>
<tr><th></th><th>lichen</th><th>bare_rocks</th><th>snow</th></tr>
<tr><td>lichen</td><td>91</td><td>0</td><td>1</td></tr>
<tr><td>bare_rocks</td><td>1</td><td>0</td><td>19</td></tr>
<tr><td>snow</td><td>0</td><td>74</td><td>0</td></tr>
</table></td><td>98.92 %</td></tr>
</table>

Fig 4.32 Resubstitution accuracy for RF algorithm (hills, 30 trees)

4.6.4 Random forest with a number of trees are 40.

<table>
<tr><th>Method</th><th>Confusion matrix</th><th>Resub-Accuracy</th></tr>
<tr><td>Number of trees are 40</td><td><table>
<tr><th></th><th>lichen</th><th>bare_rocks</th><th>snow</th></tr>
<tr><td>lichen</td><td>91</td><td>0</td><td>1</td></tr>
<tr><td>bare_rocks</td><td>0</td><td>1</td><td>19</td></tr>
<tr><td>snow</td><td>0</td><td>74</td><td>0</td></tr>
</table></td><td>98.92 %</td></tr>
</table>

Fig 4.33 Resubstitution accuracy for RF algorithm (hills, 40 trees)

Here, for every pixel, the model predicts the class id. Let's say 60% of a pixel is bare rock and 40% of it is covered by lichen, then the model would associate class id ‘2’ with that pixel. Now, if that happens, we miss out on a coordinate that has a significant lichen cover. To account for such pixels instead of associating a label with every pixel we need to generate confidence values. Using confidence values, not only do we not miss out on potential lichen pixels, but we would also be able to segregate pixels using the desired threshold. This can be achieved using ‘set_output_mode’ in GEE.

GEE supports probabilistic outputs when the target variables assume only two values. So, we merged snow and bare_rocks feature collections into one and used lichen as the second class.

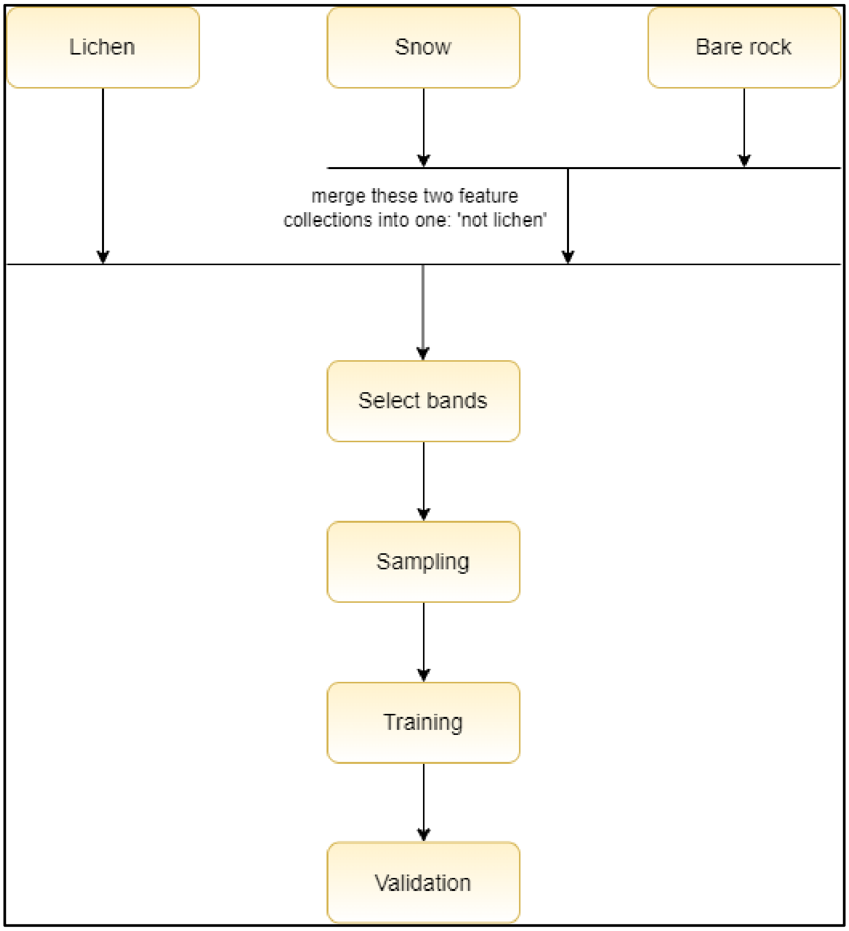


Fig 4.34 Model training with probability

Total number of training pixels:

Lichens	92	not_lichens	94

4.6.5 Random forest with a number of trees are 10 (with probability).

<table>
<tr><th>Method</th><th>Confusion matrix</th><th>Resub-Accuracy</th></tr>
<tr><td>Number of trees are 10</td><td><table><tr><td></td><td>lichen</td><td>not_lichen</td></tr><tr><td>lichen</td><td>92</td><td>0</td></tr><tr><td>not_lichen</td><td>3</td><td>91</td></tr></table></td><td>98.31 %</td></tr>
</table>

Fig 4.35 Resubstitution accuracy for RF algorithm (hills, 10 trees)

4.6.6 Random forest with a number of trees are 20. (with probability).

<table>
<tr><th>Method</th><th>Confusion matrix</th><th>Resub-Accuracy</th></tr>
<tr><td>Number of trees are 20</td><td><table><tr><td></td><td>lichen</td><td>not_lichen</td></tr><tr><td>lichen</td><td>92</td><td>0</td></tr><tr><td>not_lichen</td><td>1</td><td>91</td></tr></table></td><td>99.50 %</td></tr>
</table>

Fig 4.36 Resubstitution accuracy for RF algorithm (hills, 20 trees)

Chapter 5

Evaluation

5.1 Testing (Larsemann Hills)

For validation, we took points from Larsemann hills which were provided by our mentor. So, our feature collection had a total of 321 points out of 232 points under 'lichen' and 69 points under 'snow', and 20 points under 'bare rock'.

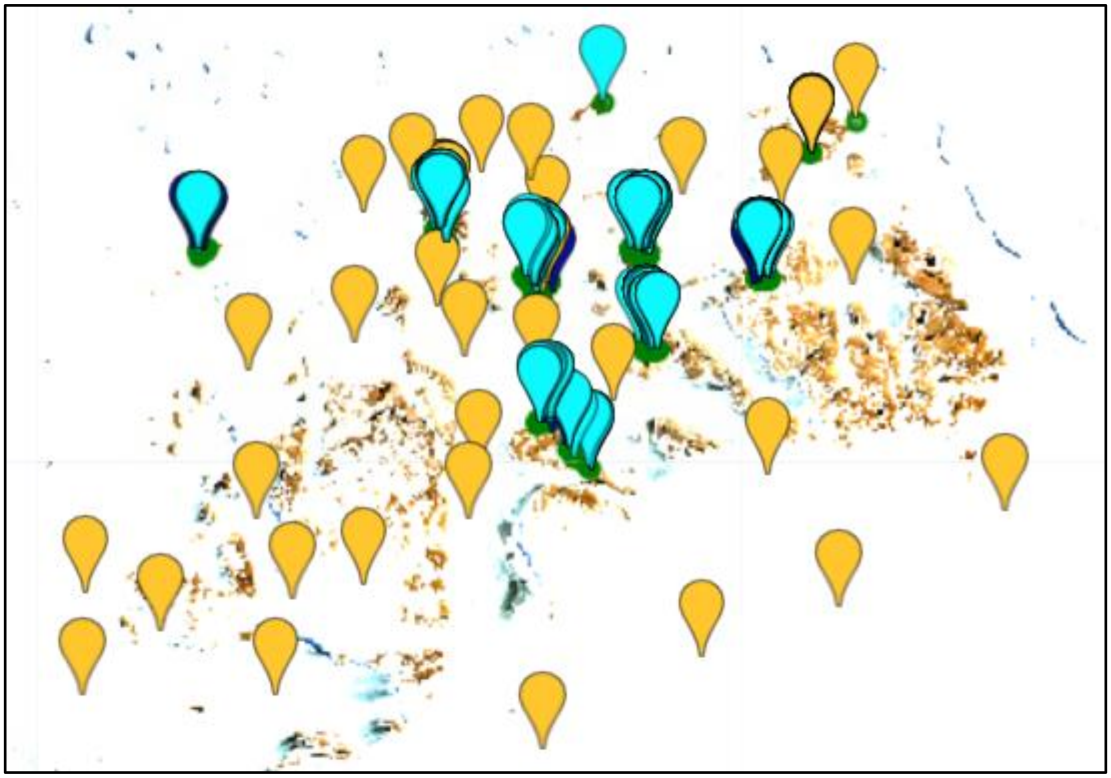


Fig 5.1 Validation points in larsemann hills

Total number of validation pixels:

Lichens	232
Bare_rocks	20
Snow	69

5.1.1 Random forest with a number of trees are 10.

<table>
<tr><th>Method</th><th>Confusion matrix</th><th>Test Accuracy</th></tr>
<tr><td>Number of trees are 10</td><td><table>
<tr><th></th><th>lichen</th><th>bare_rocks</th><th>snow</th></tr>
<tr><td>lichen</td><td>182</td><td>14</td><td>36</td></tr>
<tr><td>bare_rocks</td><td>4</td><td>63</td><td>0</td></tr>
<tr><td>snow</td><td>4</td><td>3</td><td>13</td></tr>
</table></td><td>80.87 %</td></tr>
</table>

Fig 5.2 Test accuracy for RF algorithm (hills, 10 trees)

Classification result.

Green color → lichens → Id = 0

Red color → bare rocks → Id = 2

Snow → white color → Id = 3

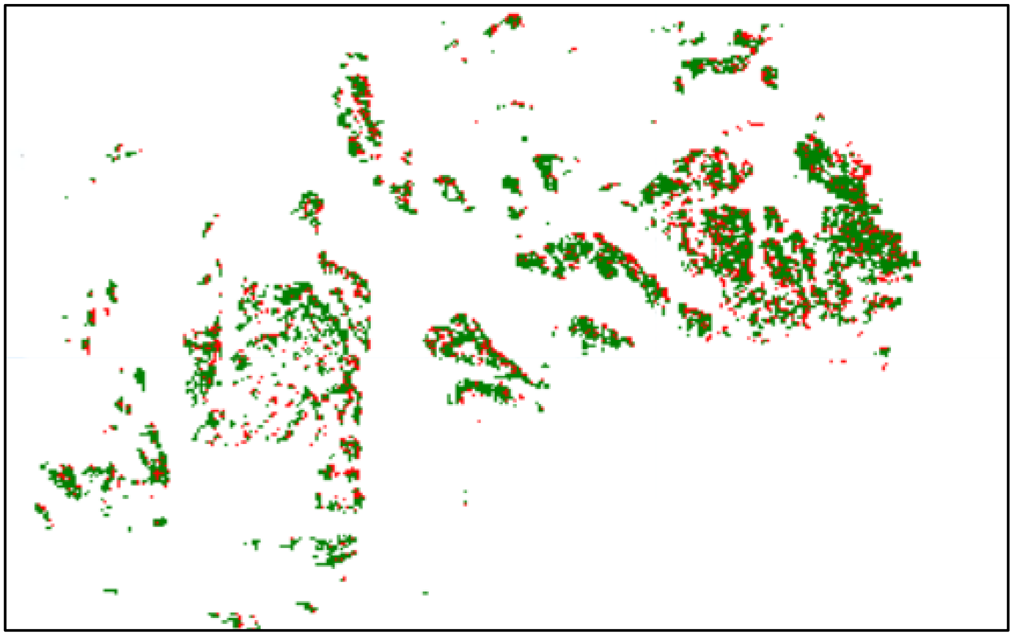


Fig 5.3 Classification of the Larsemann hills (10 trees)

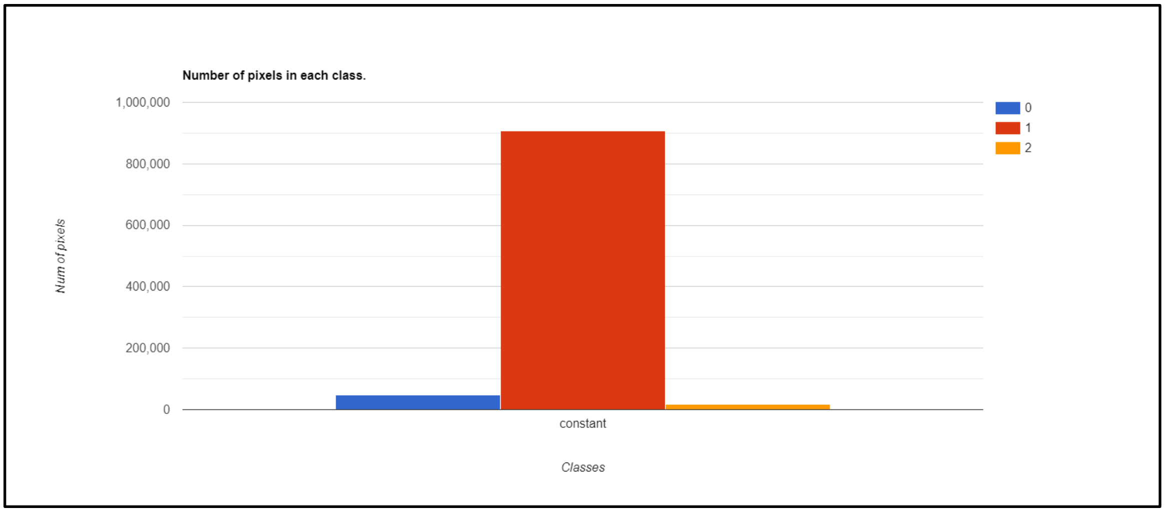


Fig 5.4 Number of pixels in each class (RF 10 number of trees).

Lichens	48,632	5.00 %
Snow	907,335	93.24 %
Bare rocks	17,133	1.76 %
Total	973,100	100 %

5.1.2 Random forest with a number of trees are 20.

<table>
<tr><th>Method</th><th>Confusion matrix</th><th>Test Accuracy</th></tr>
<tr><td>Number of trees are 20</td><td>
<table>
<tr><th></th><th>lichen</th><th>bare_rocks</th><th>snow</th></tr>
<tr><td>lichen</td><td>194</td><td>12</td><td>26</td></tr>
<tr><td>bare_rocks</td><td>4</td><td>63</td><td>0</td></tr>
<tr><td>snow</td><td>1</td><td>2</td><td>17</td></tr>
</table>
</td><td>85.89 %</td></tr>
</table>

Fig 5.5 Test accuracy for RF algorithm (hills, 20 trees)

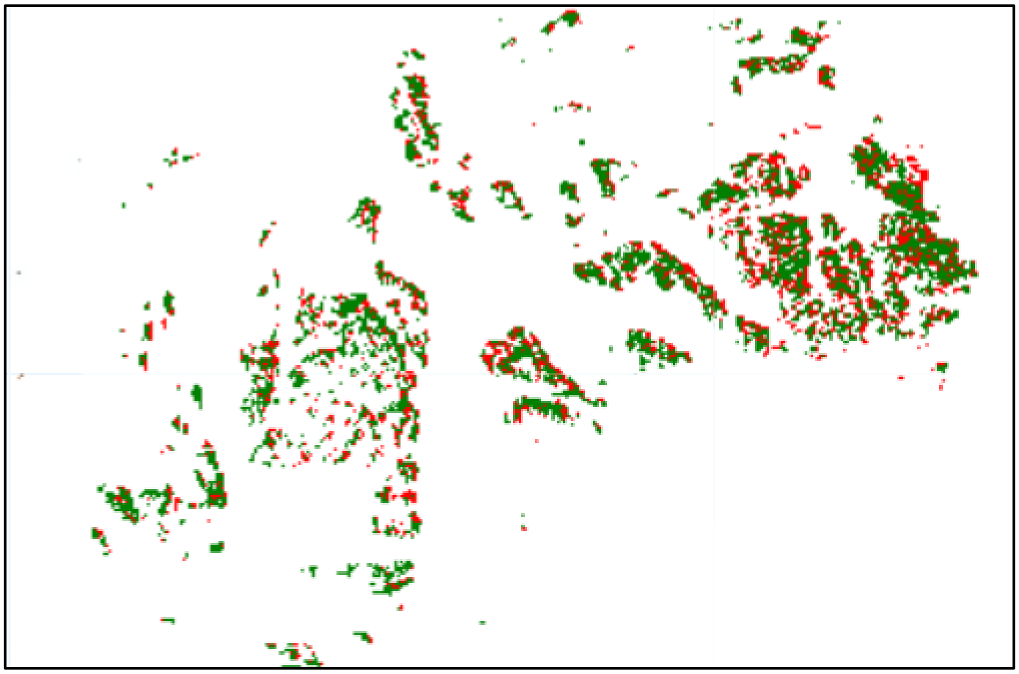


Fig 5.6 Classification of the Larsemann hills (20 trees)

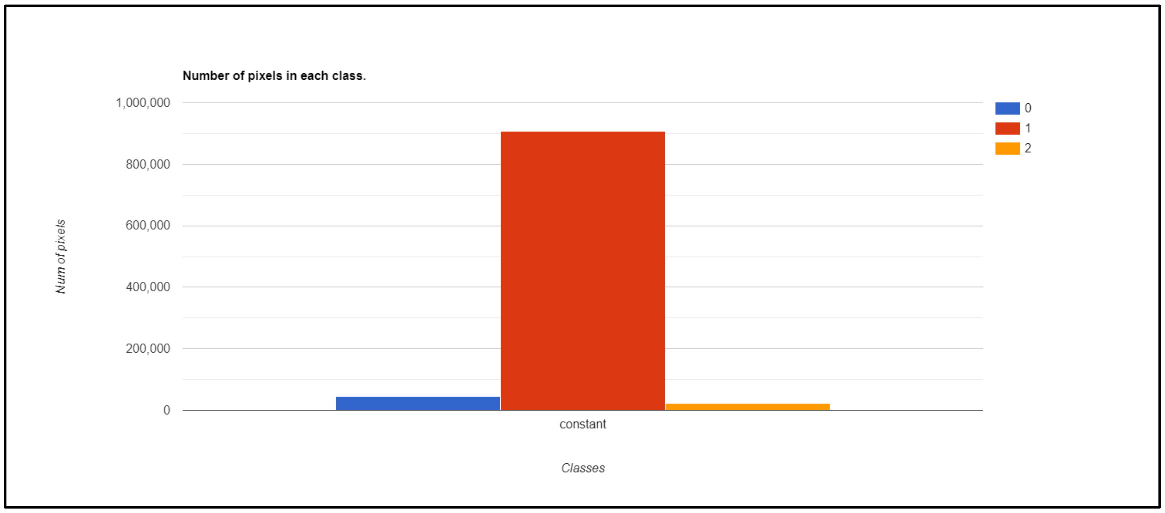


Fig 5.7 Number of pixels in each classes (RF 20 number of trees).

Lichens	44,932	4.61 %
Snow	906,073	93.11 %
Bare rocks	22,095	2.27 %
Total	973,100	100%

5.1.3 Random forest with a number of trees are 30.

<table>
<tr><th>Method</th><th>Confusion matrix</th><th>Test Accuracy</th></tr>
<tr><td>Number of trees are 30</td><td>
<table>
<tr><th></th><th>lichen</th><th>bare_rocks</th><th>snow</th></tr>
<tr><td>lichen</td><td>171</td><td>12</td><td>49</td></tr>
<tr><td>bare_rocks</td><td>4</td><td>63</td><td>0</td></tr>
<tr><td>snow</td><td>1</td><td>0</td><td>19</td></tr>
</table>
</td><td>79.31 %</td></tr>
</table>

Fig 5.8 Test accuracy for RF algorithm (hills, 30 trees)

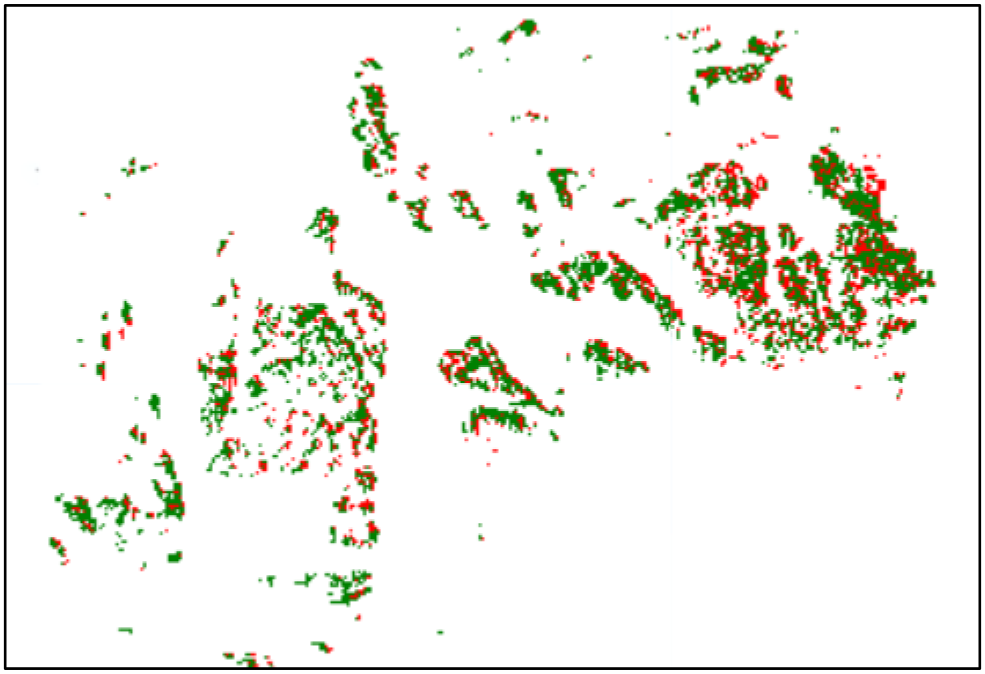


Fig 5.9 Classification of the Larsemann hills (30 trees)

Lichens	45,772	4.70 %
Snow	906,415	93.14 %
Bare rocks	20,913	2.14%
Total	973,100	100 %

5.1.4 Random forest with a number of trees are 10. (lichen and not lichen)

<table>
<tr><th>Method</th><th>Confusion matrix</th><th>Test Accuracy</th></tr>
<tr><td>Number of trees are 10</td><td><table><tr><th></th><th>lichen</th><th>not_lichen</th></tr><tr><td>lichen</td><td>183</td><td>49</td></tr><tr><td>not_lichen</td><td>4</td><td>83</td></tr></table></td><td>83.38 %</td></tr>
</table>

Fig 5.10 Test accuracy for RF algorithm (hills, 10 trees)

5.1.5 Confidence value more than 0.4.

White pixels → lichen with more than 0.4 confidence.

Black pixels → not lichen.

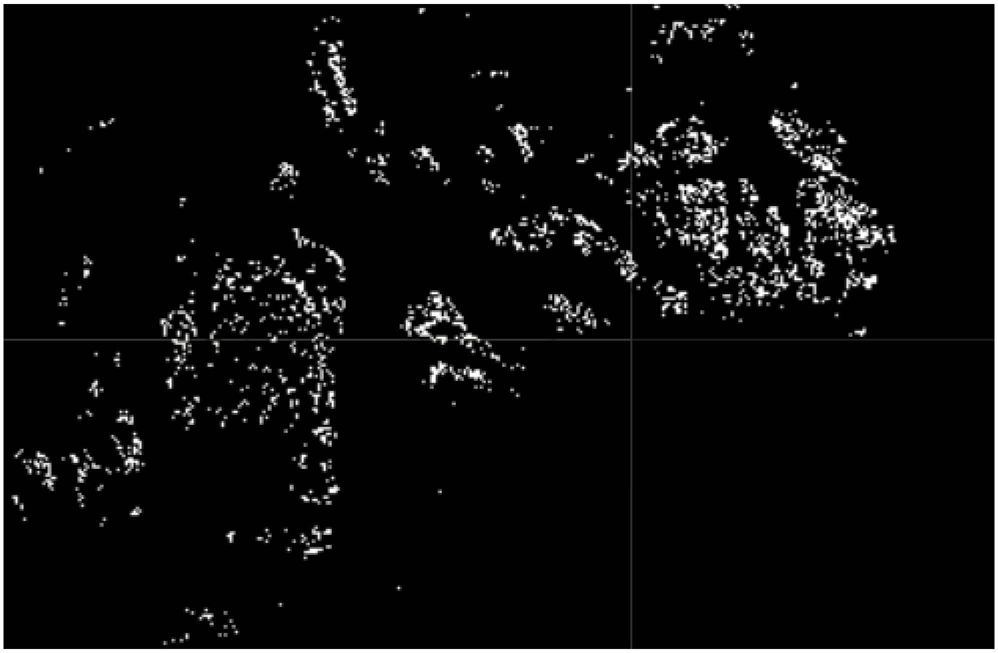


Fig 5.11 Result with confidence value more than 0.4

5.1.6 Confidence value more than 0.6.

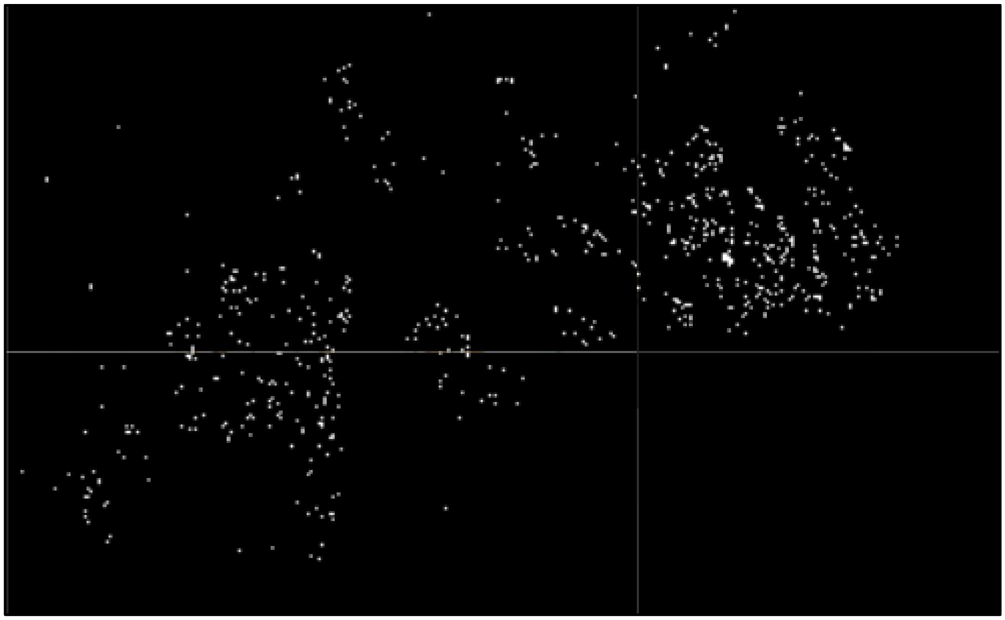


Fig 5.12 Result with confidence value more than 0.6

5.1.7 Probability in each pixel of larsemann hills islands.

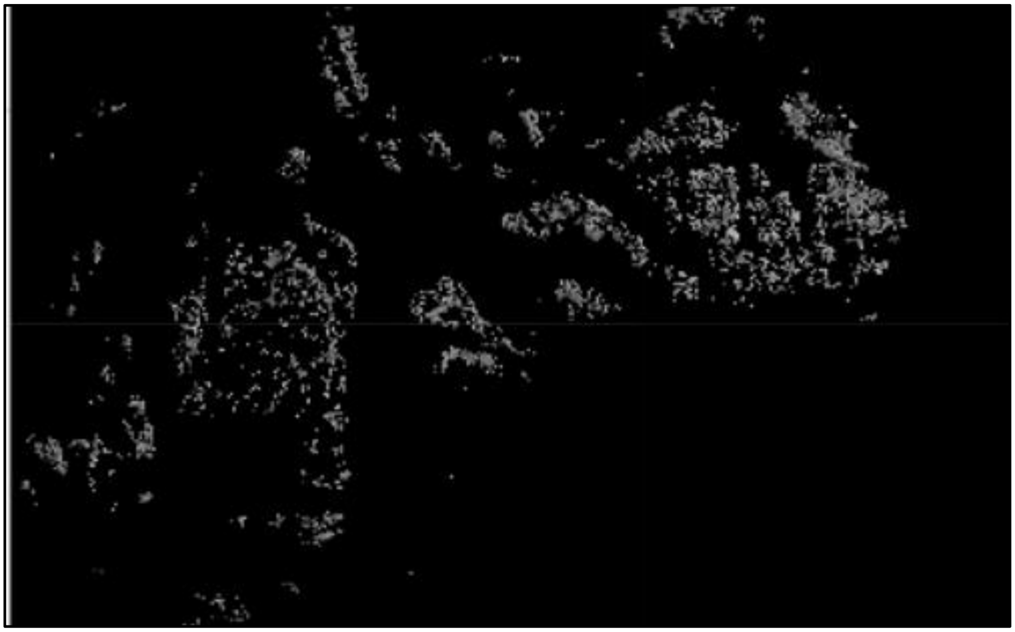


Fig 5.13 Output with probability

Chapter 6

Conclusion and Future Extension

6.1 Conclusion

Detailed comparison of Test results have been discussed above. The Following table lists test accuracy for algorithms, and the number of decision trees used in the training phase. We conclude that for bi-class classification, the Random Forest algorithm with 10 decision trees gives the best test accuracy of 83.38%, and for multi-class classification, Random Forest with 20 decision trees gives the best test accuracy of 85.89%.

Algorithm	**Number of trees**	**Number of classes**	**Test Accuracy**
Random Forest	10	3(Snow, Lichen, Bare Rocks)	80.87 %
Random Forest	20	3(Snow, Lichen, Bare Rocks)	85.89 %
Random Forest	30	3(Snow, Lichen, Bare Rocks)	79.31 %
Random Forest	10	2(Lichen, Not Lichen)	83.38 %
CART(Classification and Regression Trees)	N.A	3(Snow, Lichen, Bare Rocks)	78.24 %

Fig 6.1 Conclusion table

6.2 Future Extension

Presently all the results i.e the GeoTiff images can be viewed by adding them as a layer to the map or can be exported to Google Drive using export.toDrive() function. We aim to create an interface in addition to said options where with a simple click the user can view the results and manipulate them using features like displaying pixels of a certain class. With these features, the user can interact easily with the model, without having to make changes to the GEE script.

Bibliography

- Sentinel-2-Mission-spectral-and-SSD-requirements-The-MSI-instrument-is-based_fig1_321214918
- wikipedia.org/wiki/Random_forest
- developers.google.com-earth-engine-datasets-catalog-COPERNICUS_S2
- machinelearningmastery.com-classification-and-regression-trees-for-machine-learning
- www.analyticsvidhya.com/blog/2021/04/
- polarresearch.net/index.php/polar/article/view/3219/pdf_48
- www.analyticssteps.com/blogs/classification-and-regression-tree-cart-algorithm
- towardsdatascience.com/gradient-boosted-decision-trees-explained-9259bd8205af

- **NDVI Analysis on Jessore region**

Following pages contain results of NDVI analysis carried out for Jeesore region:

The aim was to try and find out coordinates which could be the possible location of Prosopis Juliflora(an Invasive Species) in the Jessore Eco-Sensitive-Zone(ESZ).

We were initially provided a few points(Latitude and Longitudes) of Prosopis. These points, if correct, would follow a set pattern an NDVI vs Time graph. We plotted the said graph for all these points and found a couple of points that deviated from the expected behavior. We removed these points and proceed with the remaining points as representatives of Prosopis.

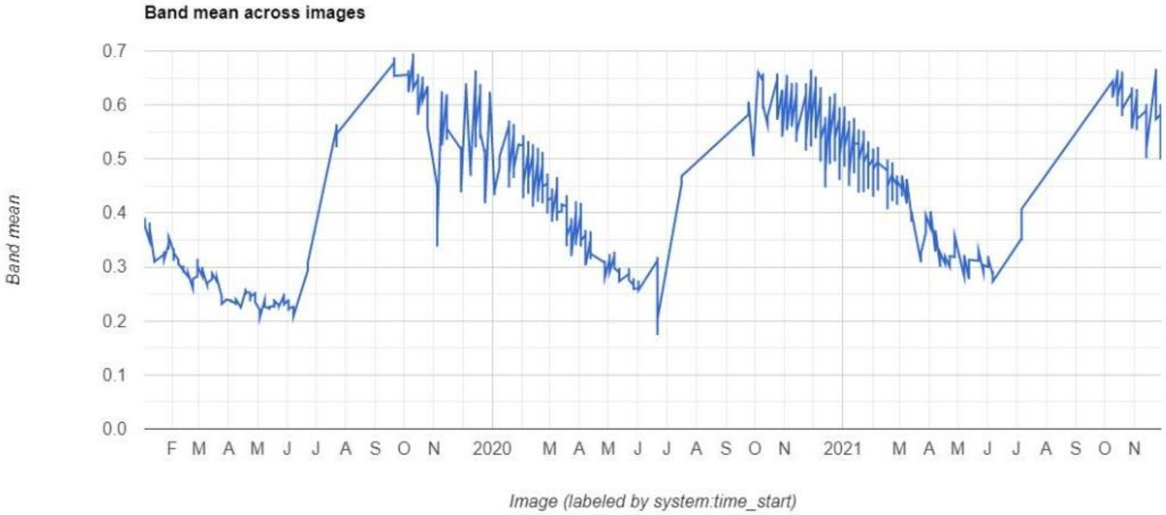


Fig 7.1 NDVI Time-series flow of 3 years

Our next task was to find out coordinates that had NDVI values of 0.3, 0.4, , and 0.5 for all years 2019, 2020, and 2021. Following are the images for 2019 and 2020 for NDVI thresholds of 0.3 - 0.4, 0.4 - 0.5, > 0.6.

Year 2020 (NDVI: 0.4-0.5):

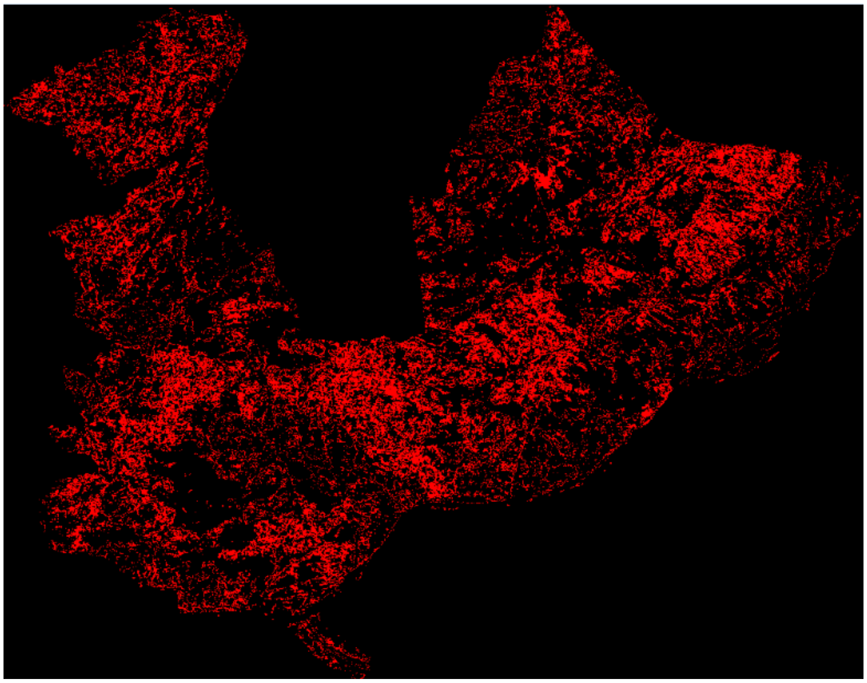


Fig 7.2 NDVI threshold 0.4 to 0.5 (2020)

Year 2020 (NDVI: 0.5-0.6):

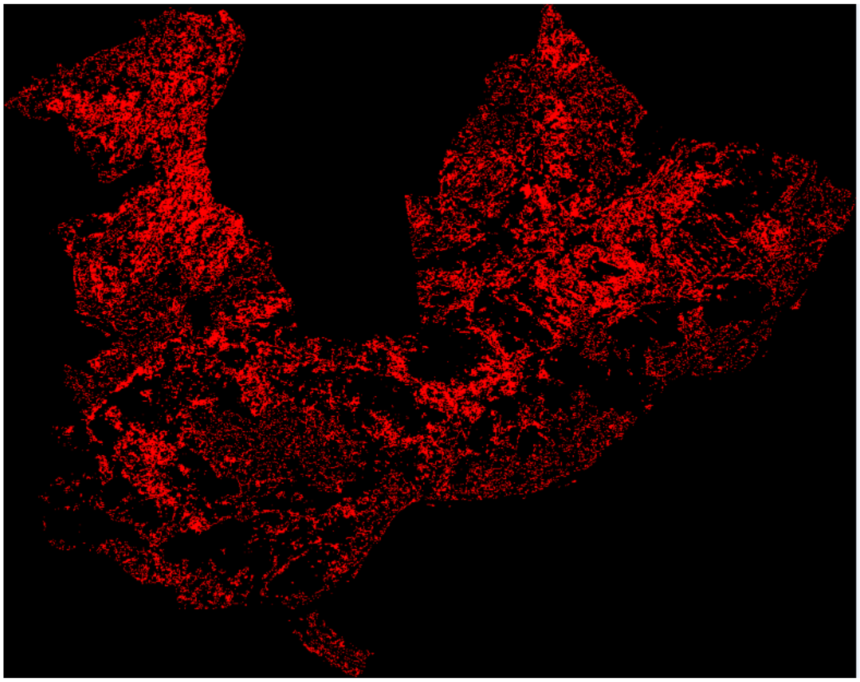


Fig 7.3 NDVI threshold 0.5 to 0.6 (2020)

Year 2020 (NDVI: 0.6+):

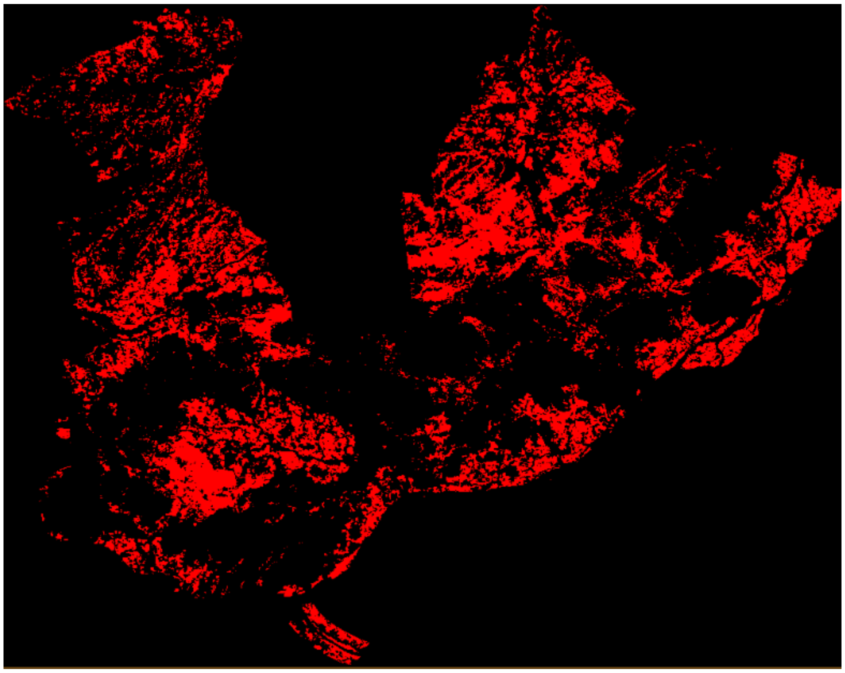


Fig 7.4 NDVI threshold more than 0.6 (2020)

Year 2021 (NDVI:0.4-0.5):

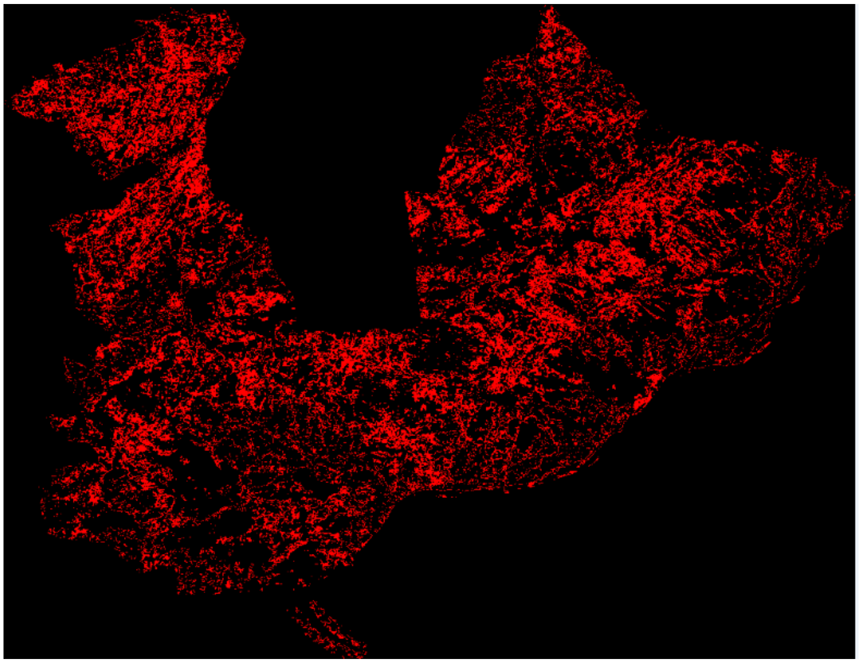


Fig 7.5 NDVI threshold 0.4 to 0.5 (2021)

Year 2021 (NDVI 0.5-0.6):

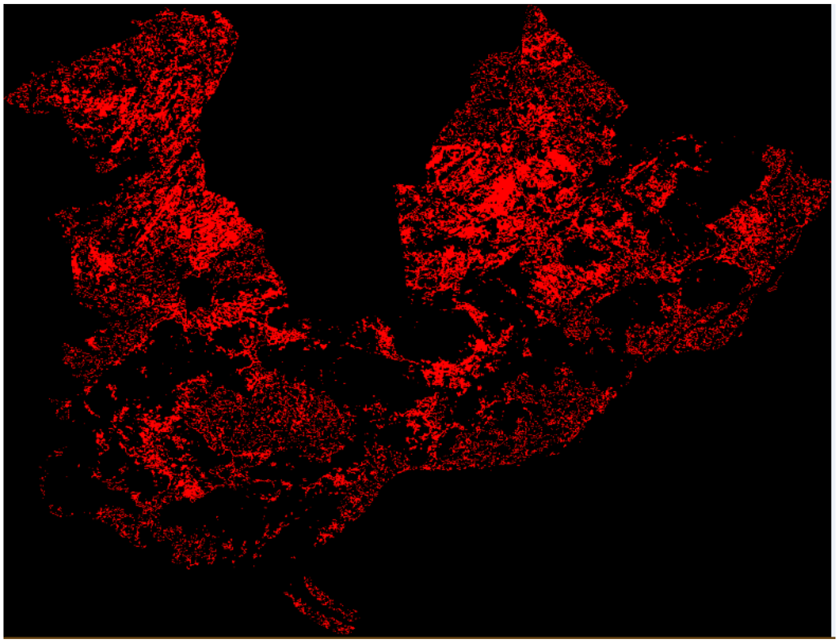


Fig 7.6 NDVI threshold 0.5 to 0.6 (2021)

Year 2021 (NDVI 0.6+):

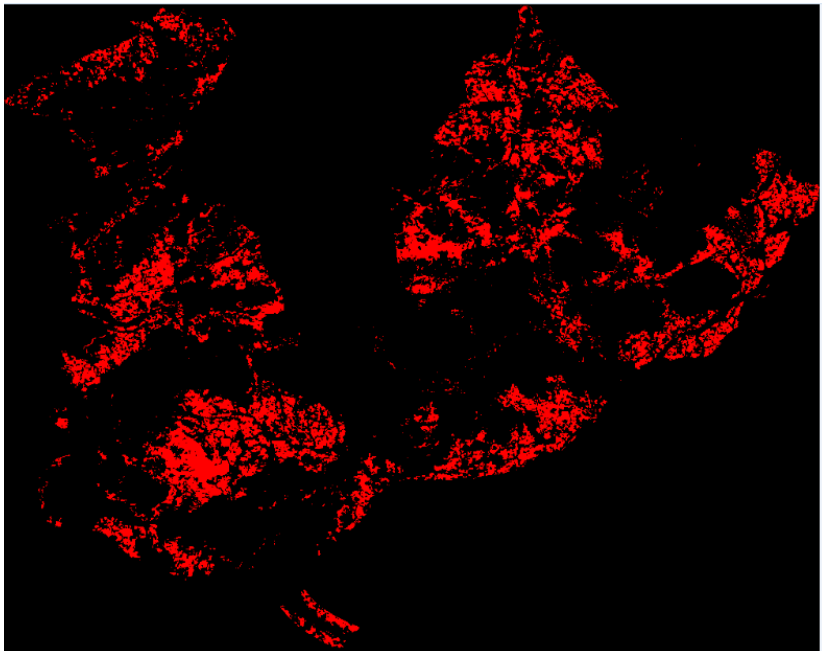


Fig 7.7 NDVI threshold more than 0.6 (2021)

By taking the intersection of 9 such geotiffs we found out coordinates that fell into a certain threshold for all the 3 years. Using these as added layers on the map, one can try and find out the threshold that best depicts the presence of Prosopis Juliflora.

For the following image, a red pixel indicates that the value of NDVI was found to be more than 0.6 for 2019, 2020, and 2021.

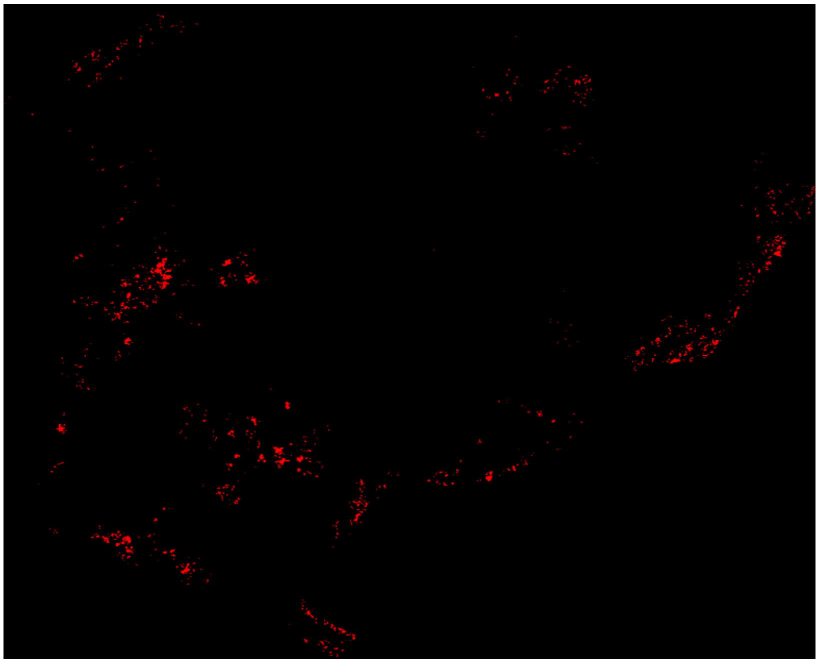


Fig 7.8 NDVI more than 0.6 (2019, 2020, 2021)

Considering only the year 2021, we can divide the region of Jessore based on a pixel's NDVI value. A dark green pixel indicates the higher value of NDVI, and a pixel a with low NDVI value would appear yellow.

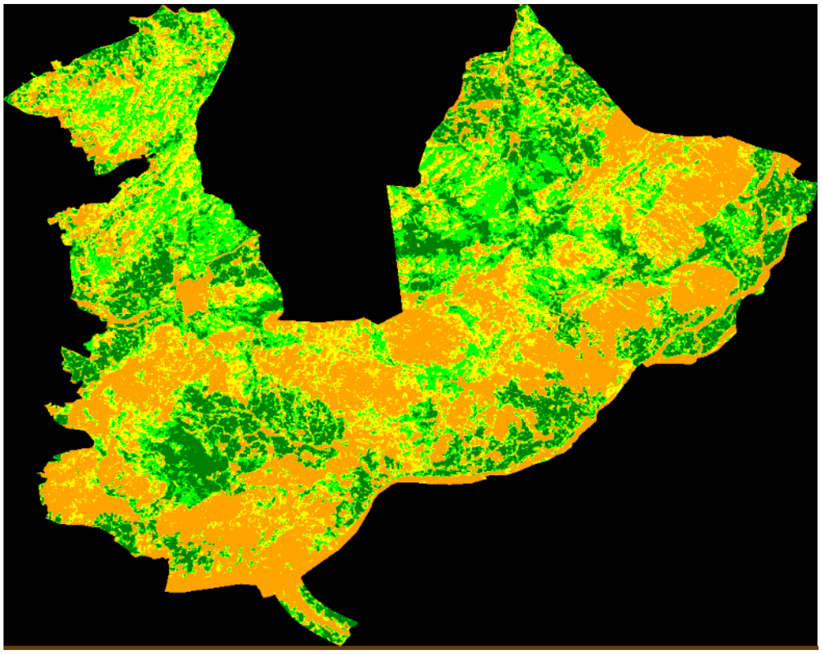


Fig 7.9 NDVI high to low pixels (dark green to yellow)

Based on these results and truth points made available to us, we trained a Random Forest Classifier to map the presence of Juliflora. We choose satellite images keeping in mind the fact that Prosopis are evergreen, whereas the native vegetation shed their leaves every year following a pattern. The following image shows in green that the model is predicted to be the coordinates of Prosopis Juliflora.

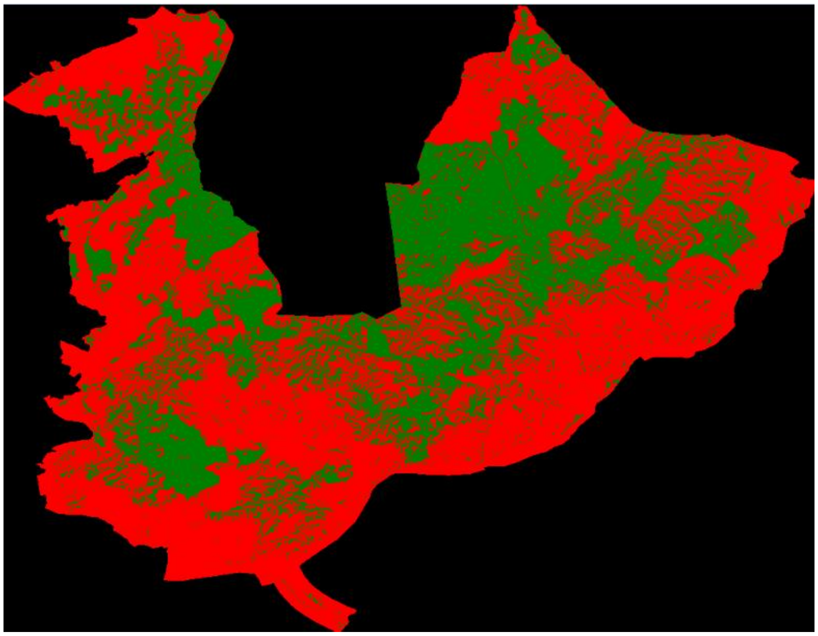


Fig 7.10 Classification of Prosopis juliflora (Jessore region)