Calculating Area of Snow Cover over Gangotri glacier using Python and Landsat 8 data from USGS

R. Virinchi I20PH008
NIT Surat

December 2023

1 Introduction

Estimating snow cover from Landsat data using the Normalized Difference Snow Index (NDSI) is a common practice in remote sensing and is often implemented using Python. The NDSI is calculated by taking the difference between the near-infrared (NIR) and shortwave infrared (SWIR) bands and dividing it by their sum. In Python, the process involves accessing Landsat data, extracting the relevant bands, and applying the NDSI formula. The resulting NDSI values can then be thresholded to identify snow-covered areas, with higher NDSI values indicating the presence of snow. This methodology allows for the efficient and accurate mapping of snow cover on the Earth's surface, facilitating applications in climate studies, water resource management, and environmental monitoring. The combination of Landsat data and Python programming provides a powerful and flexible toolset for researchers and practitioners engaged in snow cover estimation and related remote sensing applications.

2 Python code

2.1 Calculating Area of Snow Cover over Gangotri Glacier

```
import rasterio
from rasterio.enums import Resampling
import numpy as np

def calculate_ndsi(band_green, band_swir,
output_dtype=np.float64):
    """
    Calculate Normalized Difference Snow Index (NDSI).
```

```
    NDSI = (Green - SWIR) / (Green + SWIR)

    Parameters:
    band_green (numpy.ndarray): Green band array.

    band_swir (numpy.ndarray): Shortwave Infrared
    (SWIR) band array.

    Returns:
        ndsi (numpy.ndarray): NDSI array.
    """
    with np.errstate(divide='ignore', invalid='ignore'):

        ndsi = np.divide((band_green - band_swir),
        (band_green +
        band_swir), out=np.zeros_like(band_green,
        dtype=output_dtype), where=(band_green +
        band_swir) != 0)

    return ndsi

def calculate_snow_cover(ndsi_threshold, ndsi_array):
    """
    Calculate snow cover area based on a given NDSI
    threshold.

    Parameters:
        ndsi_threshold (float): NDSI threshold for snow cover.
        ndsi_array (numpy.ndarray): NDSI array.

    Returns:
        snow_cover_area (float): Snow cover area in
        square units.
    """
    snow_cover_pixels = np.count_nonzero(ndsi_array >
    ndsi_threshold)
    snow_cover_area = snow_cover_pixels * pixel_area  #
    Assuming pixel area is constant
    return snow_cover_area/1000000

tif_path_green = "D:\\pythonenvi\\sep
2023\\LC08_L1TP_145039_20230923_20231002_02_T1_B3.TIF"
tif_path_swir = "D:\\pythonenvi\\sep
2023\\LC08_L1TP_145039_20230923_20231002_02_T1_B6.TIF"

# Open the TIFF files using rasterio
```

```
with rasterio.open(tif_path_green) as src_green,
rasterio.open(tif_path_swir) as src_swir:
    # Read the raster bands as numpy arrays
    green_band = src_green.read(1)
    swir_band = src_swir.read(1)

    # Calculate NDSI
    ndsi = calculate_ndsi(green_band, swir_band)

    # Define NDSI threshold for snow cover
    ndsi_threshold = 0.2

    # Calculate snow cover area
    pixel_area = src_green.res[0] * src_green.res[1]  #
    Assuming
    square pixels
    snow_cover_area =
    calculate_snow_cover(ndsi_threshold, ndsi)

    print(f"Snow cover area: {snow_cover_area:.2f}
    square km")
```

Output:

```
#2014
Snow cover area: 22735.26 square km

#2017
Snow cover area: 26497.26 square km

#2020
Snow cover area: 34669.07 square km

#2023
Snow cover area: 26540.96 square km
```

2.2 Plotting the data

```
import matplotlib.pyplot as plt
x=[2014,2017,2020,2023]
y=[22735.26, 26497.26, 34669.07, 26540.96]
plt.figure()
plt.scatter(x, y, color='red')
plt.plot(x,y)
plt.title('Snow cover over the years')
plt.xlabel('year')
```

```
plt.ylabel('area in sq km')
plt.show()
```

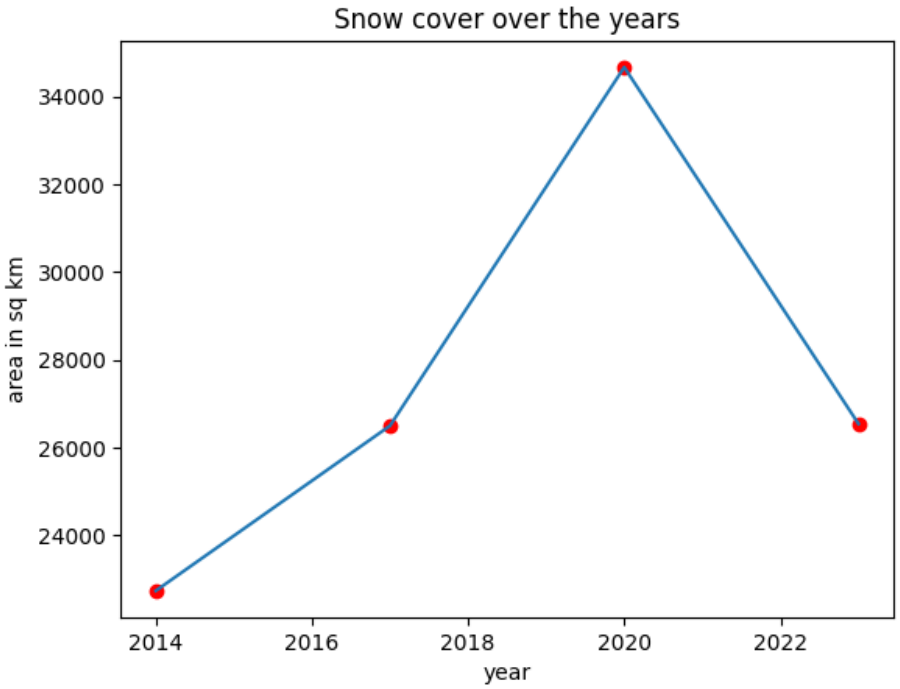

Figure 1: Year vs Area

3 Conclusion

Gangotri glacier saw an increase in area of snow cover from 2014 to 2020 and then saw a decline from 2020 to 2023.