Assignment 1

FYS-3023

Trym Varland
GitHub: tva050

UiT - The Artic University of Norway
Tromsø - Norway

Task 1

a)

In Figure 1 we have displayed both the VV-channel and the VH-channel from the pre-fire S1 file. We can see in the image for the VV-channel, that water areas is easy to identify as in the image it appears darker, we some lakes and rivers. We are also able to see differences in vegetation, as we see some areas in the image appear darker or lighter but you need to look closely, darker areas corresponds to more dens vegetation. In the context of *SAR* ambiguities, we may be able to identify shadowing as we can see some darker lines which are areas behind tall object, where little to no radar signal is returned. In the context of speckle noise, I cannot identify anything directly by looking at the images but when they are open in *SNAP* it appears speckle noise on the images.

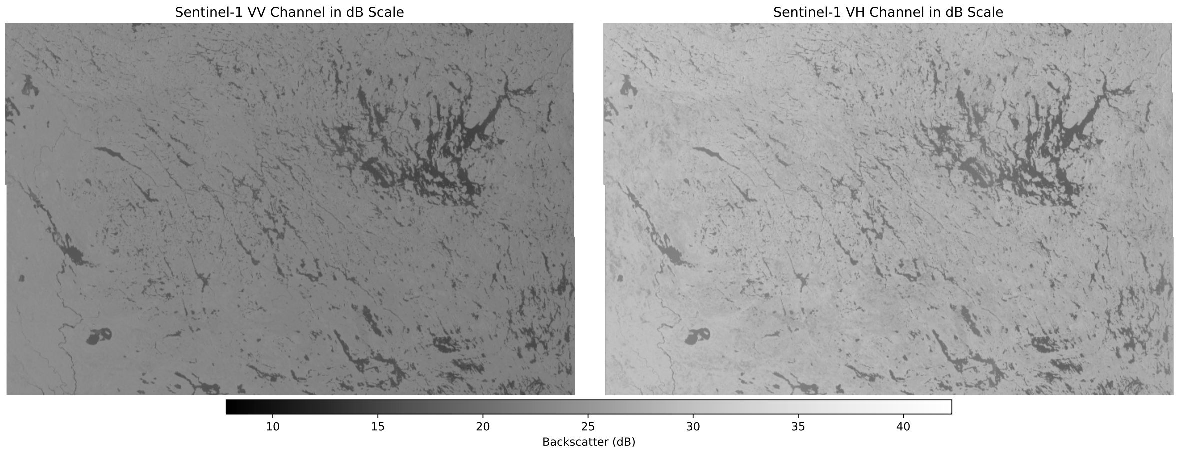


Figure 1: Left is the VV-channel and to the right is the VH-channel of the pre-fire S1 file.

b)

acquisition time

- 29-MAY-2023 22:53:34.639191 utc

geographical extent

- Top left corner: lat 47deg44'34"N, lon 77deg06'43"W
- Bottom left corner: lat 49deg14'20"N, lon 77deg33'38"W
- Top right corner: lat 48deg07'53"N, lon 73deg47'43"W
- Bottom right corner: lat 49deg37'43"N, lon 74deg08'21"W

spatial resolution

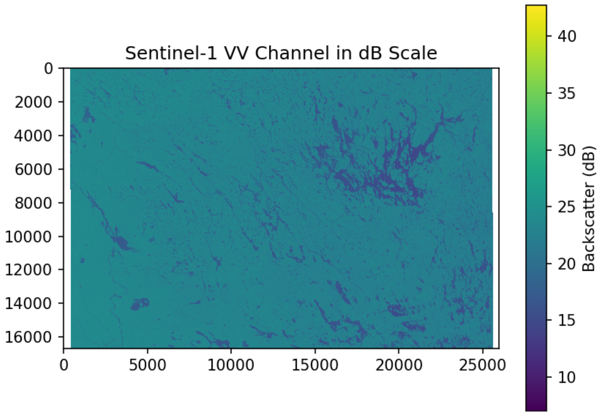
- 10x10

c)

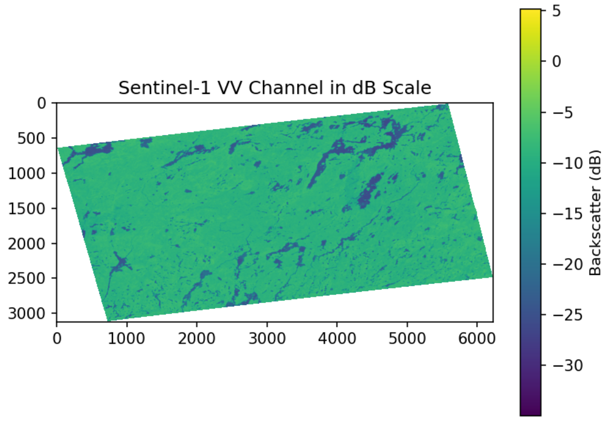
Using SNAP, the following pre-processing methods were used.

1. Subsetting
2. Thermal noise removal
3. Calibration
4. Radiometric terrain flatting
5. Speckle filtering
6. Multilooking
7. Range-Doppler terrain correction

An unprotected SAR image is sometimes "upside down", this depends on if the image is an ascending or a descending image. The image we are working with is an ascending image, hence the satellite is moving northward during the acquisition.
Thermal noise comes from the radar sensor itself rather than the Earth's surface, so it needs to be removed to enhance the quality of backscatter signals
When we do the calibration process, it has the purpose of converting the pixel values from digital numbers to radar backscatter values (e.g. Beta0 or Sigma0). Beta0 is the radar backscatter normalized, to the incident angle, this is useful for flat surfaces (e.g. calm water) or where we have large areas with similar surfaces. Sigma0 is the normalized radar cross-section, but Beta0 is more appropriate when we are interested in the surface roughness as we are in this case.
The parameter of choice when speckle filtering, was to use the Refined Lee filter as it is often recommended as it balances noise reduction and preserving edges [1]. Speckle is often called salt-and-pepper noise or granular noise.
When multilooking, we want to reduce speckle by averaging multiple looks, but this comes at the cost of reduced spatial resolution. Here we choose 3 looks as it is a decent amount for averaging, without using the overdo the computational power.
Image coordinates refer to the pixel coordinates within the SAR image, while the geo coordinates refer to the geographical coordinates (latitude and longitude) on earth. Image coordinates are useful when working with the original SAR geometry for feature extraction and object recognition. The geo-coordinates are better when comparing SAR data with other geo data e.g. optical imaging or doing further geographical analysis. The advantage of picking a specific system is it making it easier to overlay with other datasets, for when e.g. conducting spatial analysis or when comparing images as in our case.
In Figure **??** shows the different between the pre-processed SAR VV channel image (to the left), and the image after it is processed. As we see, it is a big difference not only that the Figure 1b) is is tilted and is a zoomed in part of image 1a) but we also have an image with less noise/speckle and a way lower decibel scale ranging from -30 to 5 now.



(a) Pre-processing



(b) After processing

d)

We started by repeating the process in task c), for the post-fire S1 image. We then ensured that the images had a common extent, so we cropped them to match the outline of the pre-fire S2 image provided. For the S2 image, we chose the 10m B2 band. The code for this can be seen in the appendix.
The difference of two values in decibels is equivalent to a ratio because:

$$\Delta\gamma^0_{XY} = 10log_{10}(\gamma^0_{XY-pre}) - 10log_{10}(\gamma^0_{XY_{post}}) \tag{1}$$

$$\Downarrow \text{simplifies} \Downarrow \tag{2}$$

$$\Delta\gamma^0_{XY} = 10log_{10}(\frac{\gamma^0_{XY-pre}}{\gamma^0_{XY_{post}}}) \tag{3}$$

In the simplified form the subtraction in the decibel scale corresponds to the logarithm of a ratio linearly. This means that the log ratio shows the relative change between two images, so in our case, we will have a difference between the pre-and post-fire images, where positive values indicate an increase in backscatter and negative values indicate a decrease.

e)

Now we are going to display the false color composite from 1d as an RGB image, this makes us obtain the image shown in Figure 3 below.

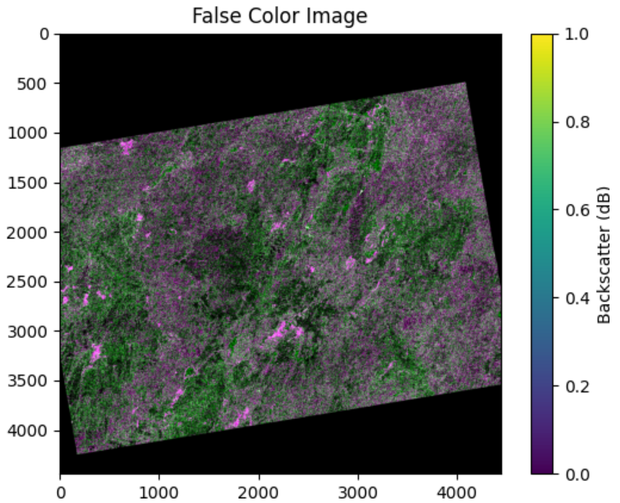


Figure 3: False color composition, with the channels: red=$\Delta\gamma_{VH}$, green=$\Delta\gamma_{VV}$, blue=$\Delta\gamma_{VH}$

f)

Now we are going to calculate the difference burn ratio (dNBR) which is an established method for detecting forest fires from optical images, and is defined as.

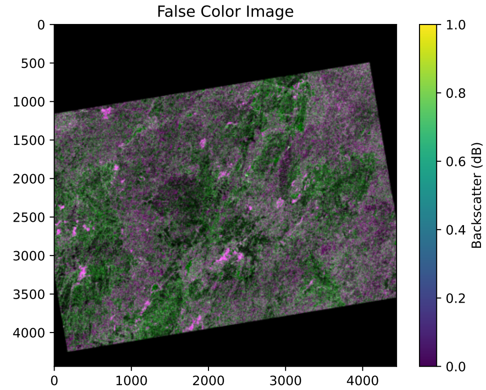
$$NRB = \frac{NIR - SWIR}{NIR + SWIR} \tag{4}$$

$$dNBR = NBR_{pre-fire} - NBR_{post-fire} \tag{5}$$

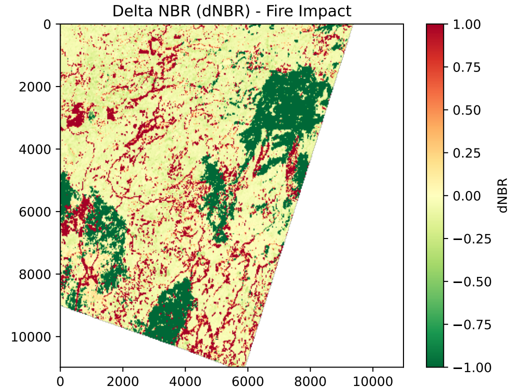
We can then implement this, to get a $dNBR$ fire impact plot. The implementation can be found in the appendix.

g)

In the Sentinel-1 RGB image, the areas of change will indicate regions where the backscatter intensity before and after the fire has changed, which can be caused by burned vegetation or structural changes. In the Sentinel-1 image, shown in Figure 4a we have the colors red, green, and blue which represent different backscatter changes. The areas of significant change will have bright colors. The change of backscatter is due to several factors, but I will mention three reasons. One is burned vegetation, fire reduces the vegetation, leading to loss of volume and surface roughness, this causes a decrease in backscatter intensity (especially for the VH channel). second is soil exposure, burned areas may expose the soil, which has a different backscatter response, than vegetation. Third is moisture change, burned areas might lose moisture content which SAR is sensitive to, and will appear as a change in backscatter.
We can see that Figure 4a with the bright pink spots correspond well with the image in Figure 4b. We can see that in the false color, we have a bright pink point at around 2000, which corresponds to what we see in the dNBR image at around 4000. Another region with appear affected by the forest fire is at (3000, 2000) in the false color image we have a darker pink strip this corresponds well with what we see in the dNBR image at (7500,4000).



(a) False color image (log-ratio) - fire impact



(b) dNBR - fire impact

Figure 4: A comparison between RGB false color (log-ratio) composition and dNBR image fire impact recognition.

The advantage of the log-ratio method is that SAR penetrates clouds and smote including that is functional for any weather conditions and at night. Another advantage is that it is sensitive to surface roughness and structural changes. A disadvantage is that the interpretation can be challenging as the backscatter can result from multiple factors, making it harder to directly attribute changes to fire damages by itself. Another disadvantage is that we have lower spatial resolution compared with Sentinel-2 optical data.
The advantage of using the dNBR (Sentinel-2 optical) method is that we get a direct vegetation analysis as we compare the near-infrared (NIR) and shortwave-infrared (SWIR) bands. We also have a higher spatial resolution than Sentinel-1 data. A disadvantage is that we are not able to penetrate clouds, smoke, or at night as it is an optical sensor.

h)

For a long-term fire monitoring program from space, I would have used thermal sensors for active fire detection rather than optical sensors for post-fire burn scar mapping and also for long-term monitoring of the ecosystem. SAR sensors will be used instead for areas with high cloud coverage or long dark periods. By combining all these sensors we will be able to have real-time fire detection, detailed post-fire analysis, and long-term ecological monitoring. We will also be able to map forest fire burn scars precisely with the thermal and optical sensors.

Appendix

```python
import os
import numpy as np
from osgeo import gdal
import matplotlib.pyplot as plt
from mpl_toolkits.axes_grid1 import make_axes_locatable


raw_s1_pre_fire_vv = r"C:\Users\trym7\OneDrive - UiT Office 365\skole\FYS-3023\Assignment 1\S1A_IW_GRDH_1SDV_20230529T225334_20230529T225359_048755_05DD08_4D82.SAFE\measurement\s1a-iw-grd-vv-20230529t225334-20230529t225359-048755-05dd08-001.tiff.ovr"
raw_s1_pre_fire_vh = r"C:\Users\trym7\OneDrive - UiT Office 365\skole\FYS-3023\Assignment 1\S1A_IW_GRDH_1SDV_20230529T225334_20230529T225359_048755_05DD08_4D82.SAFE\measurement\s1a-iw-grd-vh-20230529t225334-20230529t225359-048755-05dd08-002.tiff.ovr"

s1_pre_fire = r"ProcessedData\Prefire_S1A_IW_GRDH_1SDV_20230529T225334_20230529T225359_048755_05
DD08_4D82_tnr_Cal_TF_Spk_ML_TC.tif"
s1_post_fire = r"ProcessedData\postfire_S1A_IW_GRDH_1SDV_20230622T225335_20230622T225400_049105_05
E7A2_5922_tnr_Cal_TF_Spk_ML_TC.tif"

s2_band_10m = r"C:\Users\trym7\OneDrive - UiT Office 365\skole\FYS-3023\Assignment 1\S2A_MSIL2A_20230531T160901_N0509_R140_T18UVV_20230601T000958.SAFE\GRANULE\L2A_T18UVV_A041464_20230531T161646\IMG_DATA\R10m\T18UVV_20230531T160901_B02_10m.jp2"

s2_pre_fire_NIR = r"C:\Users\trym7\OneDrive - UiT Office 365\skole\FYS-3023\Assignment 1\S2A_MSIL2A_20230531T160901_N0509_R140_T18UVV_20230601T000958.SAFE\GRANULE\L2A_T18UVV_A041464_20230531T161646\IMG_DATA\R10m\T18UVV_20230531T160901_B08_10m.jp2"
s2_pre_fire_SWIR = r"C:\Users\trym7\OneDrive - UiT Office 365\skole\FYS-3023\Assignment 1\S2A_MSIL2A_20230531T160901_N0509_R140_T18UVV_20230601T000958.SAFE\GRANULE\L2A_T18UVV_A041464_20230531T161646\IMG_DATA\R20m\T18UVV_20230531T160901_B12_20m.jp2"

s2_post_fire_NIR = r"C:\Users\trym7\OneDrive - UiT Office 365\skole\FYS-3023\Assignment 1\S2A_MSIL2A_20230620T160901_N0509_R140_T18UVV_20230620T234755.SAFE\GRANULE\L2A_T18UVV_A041750_20230620T160904\IMG_DATA\R10m\T18UVV_20230620T160901_B08_10m.jp2"
s2_post_fire_SWIR = r"C:\Users\trym7\OneDrive - UiT Office 365\skole\FYS-3023\Assignment 1\S2A_MSIL2A_20230620T160901_N0509_R140_T18UVV_20230620T234755.SAFE\GRANULE\L2A_T18UVV_A041750_20230620T160904\IMG_DATA\R20m\T18UVV_20230620T160901_B12_20m.jp2"


def task_a():
    vh_data = gdal.Open(raw_s1_pre_fire_vh).ReadAsArray()
```

```
    vv_data = gdal.Open(raw_s1_pre_fire_vv).ReadAsArray()

    vh_data_db = 10 * np.log10(vh_data)
    vv_data_db = 10 * np.log10(vv_data)

    # plot vv channel and vh channel side by side
    fig, axs = plt.subplots(1, 2, figsize=(15, 5))
    im1 = axs[0].imshow(vv_data_db, cmap='Greys_r')
    axs[0].set_title('Sentinel-1 VV Channel in dB Scale')
    axs[0].set_xlabel('Range')
    axs[0].set_ylabel('Azimuth')
    axs[0].set_aspect('auto')
    axs[0].axis('off')

    im2 = axs[1].imshow(vh_data_db, cmap='Greys_r')
    axs[1].set_title('Sentinel-1 VH Channel in dB Scale')
    axs[1].set_xlabel('Range')
    axs[1].set_ylabel('Azimuth')
    axs[1].set_aspect('auto')
    axs[1].axis('off')

    # add color bar to the left of the plots
    fig.subplots_adjust(bottom=0.2)  # Adjust bottom to make room for the colorbar
    cbar_ax = fig.add_axes([0.2, 0.1, 0.6, 0.03])  # [left, bottom, width, height] for colorbar
    cbar = fig.colorbar(im1, cax=cbar_ax, orientation='horizontal')
    cbar.set_label('Backscatter (dB)')

    plt.tight_layout()
    plt.show()



def task_c():
    # Read the pre-fire Sentinel-1 processed image and plot it
    image = gdal.Open(s1_pre_fire).ReadAsArray()
    vv_prefire = image[0]
    print(image.shape)
    vv_data_db = 10 * np.log10(vv_prefire)

    plt.imshow(vv_data_db, cmap='viridis')
    plt.colorbar(label='Backscatter (dB)')
    plt.title('Sentinel-1 VV Channel in dB Scale before fire')
    plt.show()

def task_d():
    global log_ratio_vv, log_ratio_vh
    s2 = gdal.Open(s2_band_10m)
    GeoTransform = s2.GetGeoTransform()

    # Get the bounding box of the Sentinel-2 image
    minx = GeoTransform[0]
    maxy = GeoTransform[3]
    maxx = minx + GeoTransform[1] * s2.RasterXSize
    miny = maxy + GeoTransform[5] * s2.RasterYSize

    # File paths for the cropped images
```

```
    image_prefire_crop = "ProcessedData/S1_cropped_Prefire.tif"
    image_postfire_crop = "ProcessedData/S1_cropped_Postfire.tif"

    """ Uncomment the following lines to crop the images """
    #print(f'gdalwarp -overwrite -s_srs EPSG:4326 -t_srs EPSG:32618 -te {minx} {miny} {maxx} {maxy} {s1_pre_fire} {image_prefire_crop}')
    #os.system(f'gdalwarp -overwrite -s_srs EPSG:4326 -t_srs EPSG:32618 -te {minx} {miny} {maxx} {maxy} {s1_pre_fire} {image_prefire_crop}')
    #os.system(f'gdalwarp -overwrite -s_srs EPSG:4326 -t_srs EPSG:32618 -te {minx} {miny} {maxx} {maxy} {s1_post_fire} {image_postfire_crop}')

    # Read the cropped images
    image_prefire_crop = gdal.Open(image_prefire_crop).ReadAsArray()
    vv_prefire_crop = image_prefire_crop[0]
    vh_prefire_crop = image_prefire_crop[1]

    image_postfire_crop = gdal.Open(image_postfire_crop).ReadAsArray()
    vv_postfire_crop = image_postfire_crop[0]
    vh_postfire_crop = image_postfire_crop[1]

    # Calculate the log ratio of the backscatter intensities
    def calculate_log_ration(pre_fire, post_fire, epsilon=1e-6):
        pre_fire = np.clip(pre_fire, epsilon, None)
        post_fire = np.clip(post_fire, epsilon, None)
        log_ratio = 10 * np.log10(post_fire) - 10 * np.log10(pre_fire)
        return log_ratio

    log_ratio_vv = calculate_log_ration(vv_prefire_crop, vv_postfire_crop)
    log_ratio_vh = calculate_log_ration(vh_prefire_crop, vh_postfire_crop)

def task_e():
    # Plot the false color image
    false_color = np.dstack((log_ratio_vh, log_ratio_vv, log_ratio_vh))
    false_color = np.nan_to_num(false_color, nan=0) # Replace NaN values with 0

    plt.imshow(false_color)
    plt.colorbar(label='Backscatter (dB)')
    plt.title(r'False Color Image')
    plt.show()

def task_f():
    pre_fire_NIR = gdal.Open(s2_pre_fire_NIR).ReadAsArray() # Band 8, 10m
    pre_fire_SWIR = gdal.Open(s2_pre_fire_SWIR).ReadAsArray() # Band 12, 20m

    post_fire_NIR = gdal.Open(s2_post_fire_NIR).ReadAsArray() # Band 8, 10m
    post_fire_SWIR = gdal.Open(s2_post_fire_SWIR).ReadAsArray() # Band 12, 20m

    # Resample Band 12 to 10m resolution
    def resample_bands(src_path, dst_path, target_resolution):
```

```python
        dst = gdal.Warp(dst_path, src_path, xRes=target_resolution, yRes=target_resolution)
        dst = None

    resample_bands(s2_pre_fire_NIR, 'ProcessedData/S2_pre_fire_NIR_resampled.tif', 10)
    resample_bands(s2_pre_fire_SWIR, 'ProcessedData/S2_pre_fire_SWIR_resampled.tif', 10)
    resample_bands(s2_post_fire_NIR, 'ProcessedData/S2_post_fire_NIR_resampled.tif', 10)
    resample_bands(s2_post_fire_SWIR, 'ProcessedData/S2_post_fire_SWIR_resampled.tif', 10)

    pre_fire_NIR = gdal.Open('ProcessedData/S2_pre_fire_NIR_resampled.tif').ReadAsArray()
    pre_fire_SWIR = gdal.Open('ProcessedData/S2_pre_fire_SWIR_resampled.tif').ReadAsArray()

    post_fire_NIR = gdal.Open('ProcessedData/S2_post_fire_NIR_resampled.tif').ReadAsArray()
    post_fire_SWIR = gdal.Open('ProcessedData/S2_post_fire_SWIR_resampled.tif').ReadAsArray()

    def calculate_NBR(NIR, SWIR):
        NBR = (NIR - SWIR) / (NIR + SWIR)
        return NBR

    NBR_pre_fire = calculate_NBR(pre_fire_NIR, pre_fire_SWIR)
    NBR_post_fire = calculate_NBR(post_fire_NIR, post_fire_SWIR)

    dNBR = NBR_pre_fire - NBR_post_fire

    plt.imshow(dNBR, cmap='RdYlGn_r', vmin=-1, vmax=1)
    plt.colorbar(label='dNBR')
    plt.title('Delta NBR (dNBR) - Fire Impact')
    plt.show()

if __name__ == "__main__":
    task_a()
    #task_c()
    #task_d()
    #task_e()
    #task_f()
```

Listing 1: Assginment 1 python code

Bibliography

[1] Aiyeola Sikiru Yommy, Rongke Liu, Wu, and Shuang. Sar image despeckling using refined lee filter. In *2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics*, volume 2, pages 260–265, 2015.