remote sensing

Article

Remote sensing wildfire burn severity in Larache province in Morocco using Sentinel-2 images and Google Earth Engine

Ivan Innocent Sekibenga [1]

[1] Kyambogo University, Department of Mathematics and Statistics; iis2@student.le.ac.uk, isekibenga@kyu.ac.ug

Abstract: In this work, the effectiveness of differenced Normalised Burn Ratio(dNBR) and delta Normalised Differenced Vegetation Index(dNDVI) indices derived from Sentinel-2 images in wildfire burn severity assessment has been compared and investigated by remote sensing a wildfire that ravaged Northern Larache in July of 2022. The pre and post-fire NBR together with the pre and post-fire NDVI were calculated and mapped. dNDVI and dNBR values were then calculated and used to create indices derived burn severity maps having burn classes namely; regrowth, unburned, low severity, moderate and high severity. Results show that dNBR is inversely proportional to dNDVI. Furthermore, dNBR is more accurate and versatile than dNDVI in classifying and mapping burn severity.

Keywords: wildfire; Sentinel-2; NBR; NDVI, dNBR; dNDVI; burn severity

1. Introduction

Globally, wildfires are serious threats to vegetated ecosystems, human lives, and property [1]. Changes in climate and anthropogenic activities are responsible for this increasing recurrence and ferociousness of wildfires [2,3]. Some of the anthropogenic causes include arson, unattended campfires, agricultural burning, power lines and discarded cigarettes. Mediterranean nation-states, like Morocco, are especially pregnable and vulnerable due to their peculiar climatic conditions, marked by hot and dry summers combined with strong winds [4]. Effective wildfire disaster mitigation and prevention strategies require a robust and fast comprehension of the extent and impact of wildfires [5]. Remote sensing has emerged as an indispensable tool in detecting and monitoring wildfires, providing timely, cheap and accurate information on the affected areas when compared to fieldwork surveys [6,7].

Larache province in Morocco just like its neighboring Mediterranean regions such as Portugal has experienced a notable increase in wildfire events in recent years, resulting in significant ecological and economic damages [8]. On 13th July 2022, several wildfires were reported to have erupted in the Ouezzane, Larache and Tetouan provinces of Morocco. Approximately 1,100 families from fifteen villages in Larache evacuated their homes and fled to safety. Four people died in the fire. Another 645 families had to evacuate from Tetouan and Taza. Due to the prevailing high temperatures of over 40 degrees Celsius caused by a heat wave that month and prevailing strong winds, the catalyzed flames spread quickly into even the forested areas consuming approximately 1,618 hectares (4000 acres) of forests.

To monitor, assess and map wildfire burn severity using satellite imagery, several remote sensing indices are used such as the normalised burn ratio (NBR), differenced normalised burn ratio (dNBR), normalised differenced vegetation index (NDVI), and delta normalised differenced vegetation index (dNDVI) [9,10]. Sentinel-2 images, with their high spatial, spectral, and temporal resolutions, have proven to be effective in monitoring

Citation: To be added by editorial staff during production.

Academic Editor: Firstname Lastname

Received: date
Revised: date
Accepted: date
Published: date

various environmental phenomena, including wildfire damage [11,12]. Google Earth Engine (GEE) is a powerful cloud-based platform that enables large-scale geospatial data analysis using satellite imagery [12,13]. However to the author's best knowledge, no study has investigated the potential of combining these indices, Sentinel-2 images, and GEE for assessing wildfire burn severity in Morocco, particularly in the July 2022 wildfire in Larache province.

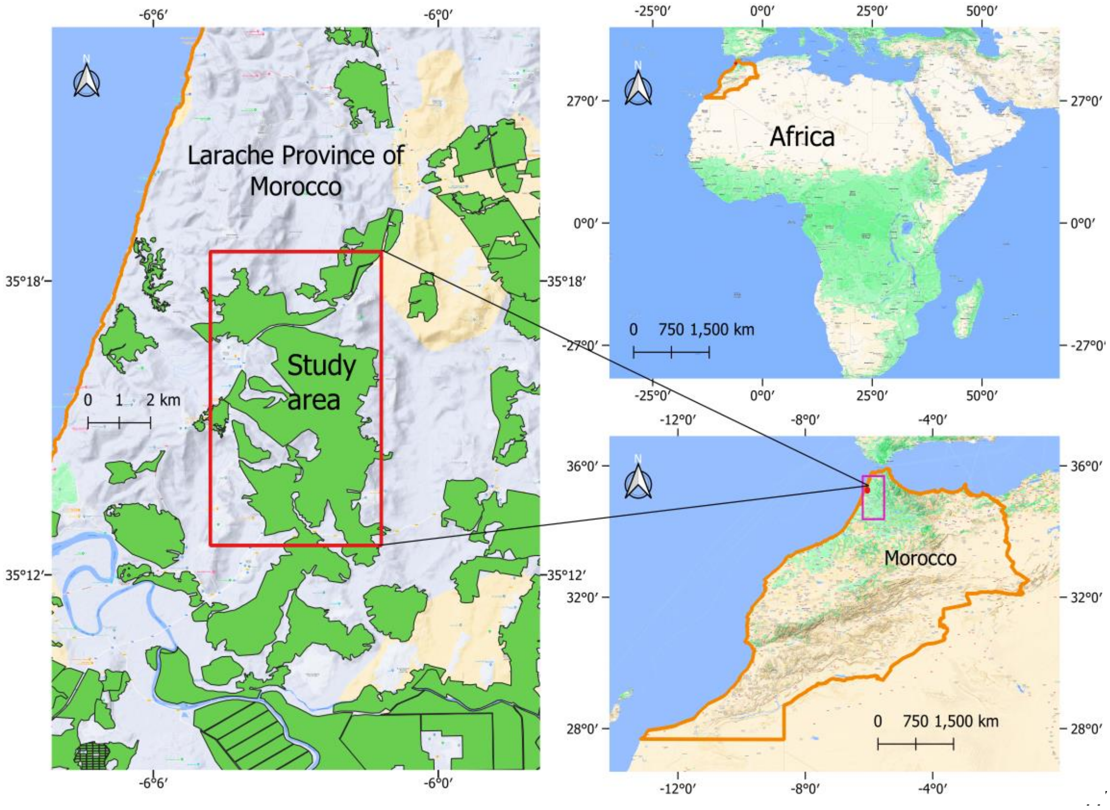
Early studies have used remote sensing techniques to detect and monitor wildfires and their effects on ecosystems [9,10]. Some researchers used moderate-resolution imaging spectroradiometer (MODIS) and Landsat imagery for wildfire detection [14,15], while others have investigated the potential of high-resolution satellite data, such as Sentinel-2, for post-fire assessment and recovery monitoring [16,17]. The integration of various indices with GEE has been explored in different environmental contexts [18,19], but its application in assessing wildfire damage, particularly in Morocco, remains limited.

This study aims to bridge this gap by employing NBR, dNBR, NDVI, dNDVI, Sentinel-2 images, and Google Earth Engine to remotely detect wildfires and burn scars in Larache province, Morocco. The study seeks to further compare the effectiveness of dNBR and dNDVI in mapping and classifying burn severity in Larache province.

2. Materials and Methods

2.1. Study Area

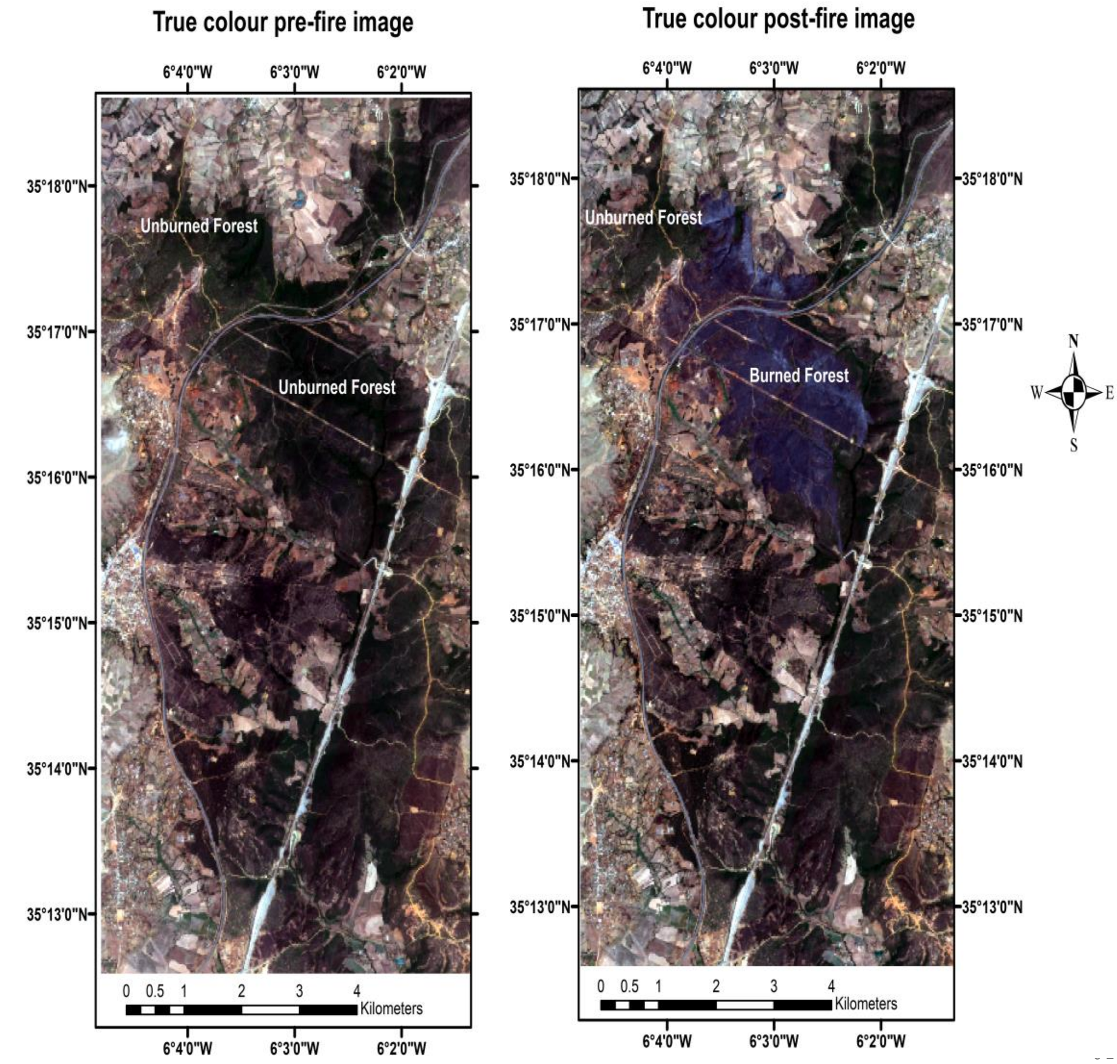
Larache province is located in Morocco (**Figure 1(a)**) on the northwestern coast at 35.1833°N 6.1500°W. Larache is its capital located at 35° 11′ 35.56" N and -6° 09′ 20.59" W. Coastal plains, mountainous areas, sandy beaches, and forests constitute its geography. Wildfires broke out on 13th July 2022 (**Figure 1(b)**) and continued burning up to the end of the month destroying over 2,330 hectares of forests (**Figure 2** and **Figure 3**).



(a)

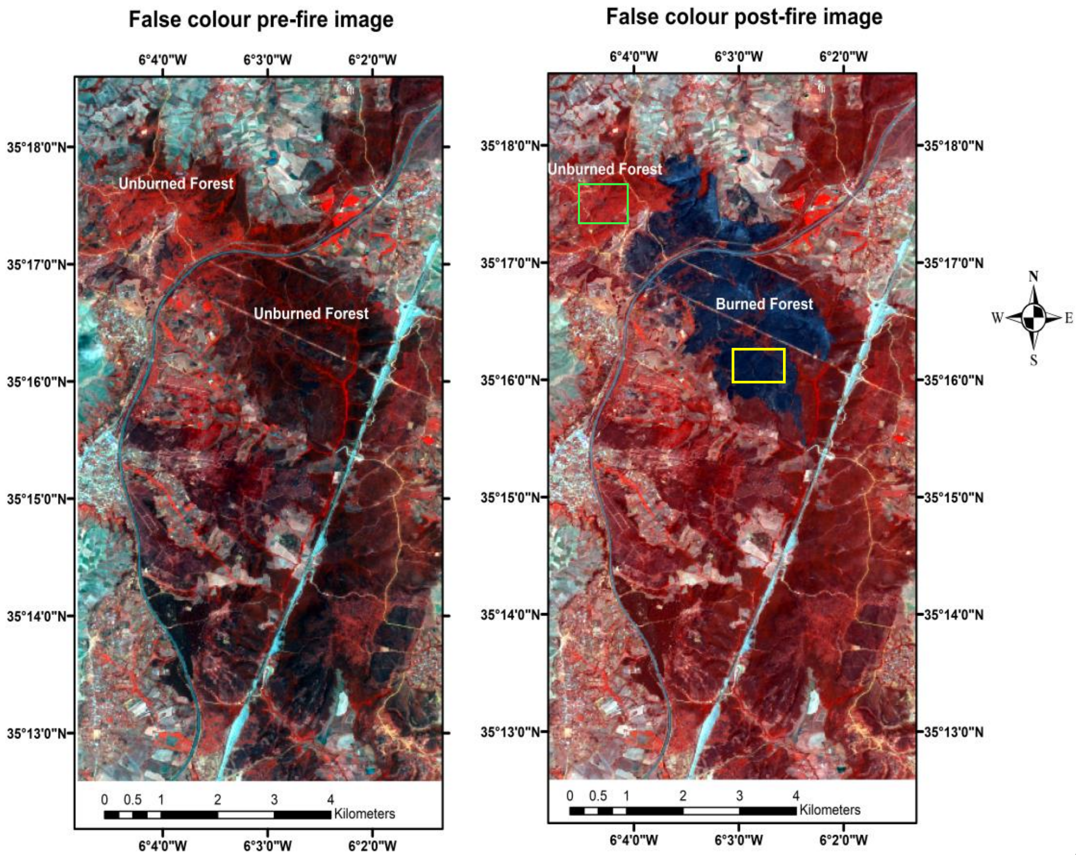
(b)

Figure 1. (**a**) Location map of Larache province in Morocco with study area indicated with red rectangle. (**b**) Photo of wildfire consuming a section of forest in Larache province, Morocco (Source:www.istockphoto.com).



(**a**) (**b**)

Figure 2. (**a**) True colour pre-fire image of Larache study area showing areas with unburned forests; (**b**) True colour post-fire image of Larache study area showing both burned and unburned forest areas.



(**a**) (**b**)

Figure 3. (**a**) False colour pre-fire image of Larache study area showing areas with unburned forests ; (**b**) False colour post-fire image of Larache study area showing areas with burned and unburned forest polygons

2.2. Remote Sensing Data

The shape file of the area of interest was downloaded from http://www.geojson.io/ with extent (-6.08, -6.02, 35.21, 35.31). This shape file was used to select the area of interest for the Sentinel-2 image downloads that followed. Two Sentinel-2 images were downloaded from the Copernicus repository using the Google Earth Engine API run using Google Colab python notebook (refer to the **Appendix**) and saved as GeoTIFF files. The following bands were downloaded and used to form image composites; Blue(B2), Green(B3), Red(B4), Near Infrared(B8) and short-wave infrared(B12). The downloaded data was at a 20m spatial resolution. Wildfires were first reported in Larache province on 13th July 2022 and some of these fires continued burning until 18th July 2022 in some areas. The date range from 1st July 2022 to 10th July 2022 was therefore used to search and obtain the pre-fire image of Larache from Google Earth Engine at a maximum cloud cover percentage of 8%. The date range from 30th July 2022 to 5th August 2022 was used to search and download the post-fire Sentinel-2 image at a maximum cloud cover percentage of 10%.

Sentinel-2 program is operated by the European Space Agency (ESA). Sentinel-2A and Sentinel-2B satellites form it having spatial resolutions of 10m for both visible and near-infrared bands. 20 m for the red-edge and shortwave infrared bands plus 60 m for the atmospheric correction band[19]. Their temporal resolution or revisit time is 5 days allowing for more frequent monitoring of changes in land cover and vegetation. Their 12-bit radiometric resolutions enable support for 4096 levels of brightness per spectral band. This high radiometric resolution helps in capturing latent fluxes in reflectance that are useful in vegetation monitoring and change detection. 13 spectral bands, ranging from visible to shortwave infrared wavelengths make Sentinel 2 images applicable to computation of vegetation indices such as Normalised Burn Ratio (NBR) and Normalised Differenced Vegetation Index (NDVI)[19].

Google Earth Engine (GEE) is a platform for robust and versatile cloud-based processing of large satellite and geospatial data. GEE can process many petabytes of data in a short time hence freeing up local resources such as computer memory and the processor. It supports scripting in Python and Java hence supporting many diverse users. It supports many algorithms such as change detection, image classification, regression and time series analysis. Visualisation of maps, images and videos is also supported [12].

2.3. Normalized Burn Ratio (NBR) and differenced Normalized Burn Ratio(dNBR)

The Normalized Burn Ratio (NBR) is a remote sensing index that detects and identifies the burned-up areas after a wildfire occurrence. Its values range from -1 to 1. NBR is calculated using the image pixel values of the Near InfraRed (NIR) and the Short-Wave InfraRed (SWIR) bands. For the sentinel-2 image, these bands are Band 8 and Band 12 respectively [9,10].

$$\mathbf{NBR} = \frac{\mathbf{NIR - SWIR}}{\mathbf{NIR + SWIR}} = \frac{\mathbf{B8 - B12}}{\mathbf{B8 + B12}} \quad (1)$$

Generally, NBR values between 0 and 1 indicate healthy green vegetation. The more positive the NBR value, the healthier and denser the vegetation cover. Negative NBR values between -1 and 0 indicate the absence of vegetation after it has been burned. The more negative the NBR value, the higher the severity of the burn scar. NBR values close to zero represent areas of negligible or non-existent change [17,20].

The differenced Normalised Burn Ratio(dNBR) is used to classify burn area severity. It is computed by taking the difference between the NBR of the pre-fire image and the NBR of the post-fire image of a study area. dNBR values range between -2 to 2[9,10]

dNBR = Pre-fire NBR – Post-fire NBR (2)

The United States Geological Survey (USGS) proposed the use of the burn severity classification below for interpretation of the severity of fire burns.

Severity level	dNBR range
Enhanced Regrowth, high(after fire)	-0.5 to -0.251
Enhanced Regrowth, low (after fire)	-0.250 to -0.101
Unburned	-0.100 to 0.99
Low severity	0.100 to 0.269
Moderate-low severity	0.270 to 0.439
Moderate-high severity	0.440 to 0.659
High severity	0.660 to 1.300

Table 1. USGS burn severity classifications using dNBR values [9].

2.3. Normalised Differenced Vegetation Index (NDVI) and dNDVI

The Normalised Differenced Vegetation Index (NDVI) is a remote sensing index that monitors the density and health of vegetation. It ranges between -1 and 1. Generally, NDVI values of -1 to 0 represent bare soil, water bodies or snow. Sparse vegetation like grasslands or low-density crops such as agricultural fields have NDVI values between 0 and 0.2. NDVI values between 0.2 and 0.5 represent moderately dense vegetation like deciduous forests, savannas or agricultural fields having medium-density crops. NDVI values over 0.5 represent dense vegetation like coniferous and rainforests or agricultural fields with high-density crops [17,20].

NDVI is generally calculated using the image pixel values of the Near InfraRed (NIR) and Red (R) bands. For the sentinel-2 image, these bands are Band 8 and Band 4 respectively.

$$\mathbf{NDVI} = \frac{\mathbf{NIR - R}}{\mathbf{NIR + R}} = \frac{\mathbf{B8 - B4}}{\mathbf{B8 + B4}} \quad (3)$$

The differenced Normalised Differenced Vegetation Index(dNDVI) is obtained by taking the difference between the NDVI of the pre-fire image and the NDVI of the post-fire image of a study area. It is a quantifier of changes in NDVI values over a period of time [17,20].

$$\text{dNDVI} = \text{Pre-fire NDVI} - \text{Post-fire NDVI} \quad (4)$$

Generally, positive dNDVI values represent gain or addition of vegetation density and growth over a period of time while negative dNDVI values represent a drop in vegetation density or growth over time. dNDVI values equal to or near zero show negligible or non-existent change in vegetation density over a period of time [20].

2.3. Methodology flow chart

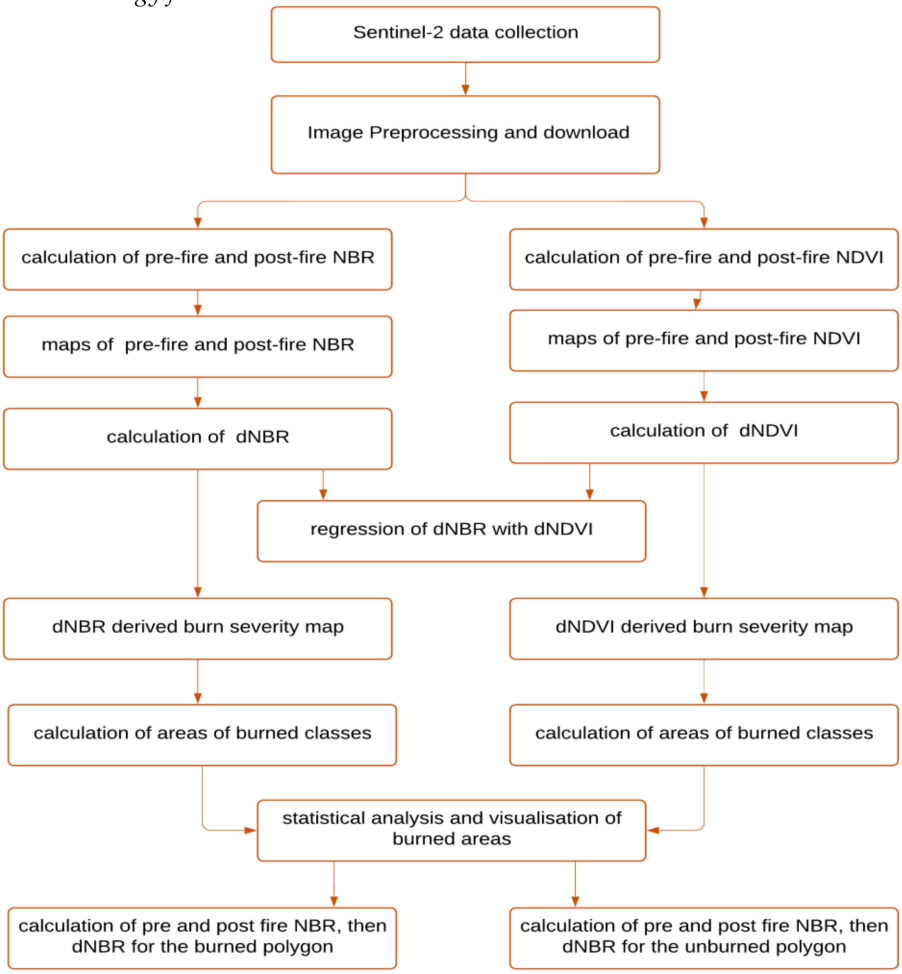


Figure 4. Flow chart showing the workflow and methodology used in this study.

3. Results

3.1. Normalised Burn Ratios for Larache

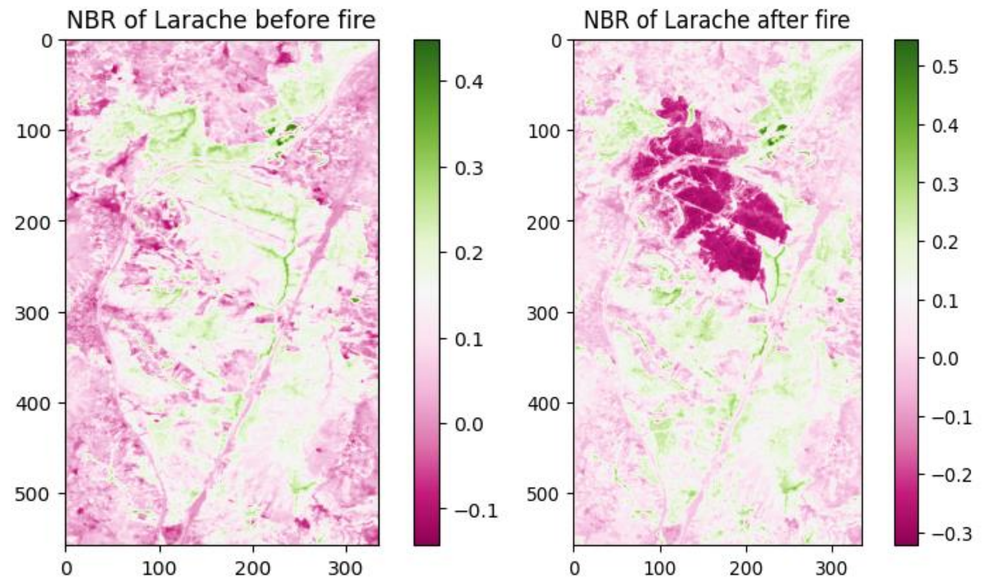
3.1.1. Normalised Burn Ratio and differenced Normalised Burn Ratio values

	Pre-fire NBR	Post-fire NBR	dNBR
Minimum	-0.143	-0.322	-0.364
Maximum	0.449	0.546	0.564
Mean	0.102	0.049	0.053

Table 2. Pre-fire and post-fire Normalized Burn Ratio values and differenced Normalised Burn Ratio values for Larache. The range of dNBR values is 0.928 and that of post-fire NBR values is 0.868

From **Table 2**, The Pre-fire image had a mean NBR value of 0.102 with a standard deviation of 0.074 and the post fire image had a mean NBR value of 0.049 with a standard deviation of 0.116. The mean decrease in the NBR of 0.053 with a standard deviation of 0.107 was indicative of the presence of a severely burned area in the post fire image.

3.1.2. Maps of Normalised Burn Ratio before and after fire



(**a**) (**b**)

Figure 5. Green colour represents areas with healthy, green vegetation while Purple colour represents areas where vegetation has been burned or removed(**a**) Pre-fire Normalised Burn Ratio map of Larache study area; (**b**) Post-fire Normalised Burn Ratio map of Larache study area.

Negative NBR values are indicative of burned up areas while positive values indicate presence of healthy vegetation. The pre-fire NBR map (**Figure 5 (a)**) shows values ranging from -0.1 to over 0.4 with few negative values which indicates absence of burned up areas but presence of healthy vegetation. The post-fire NBR map (**Figure 5 (b)**) shows values ranging from -0.3 to over 0.5 with increased presence of negative NBR values indicating presence of both a burned-up area and areas with healthy vegetation.

3.1.3. Map of differenced Normalised Burn Ratio(dNBR) with severity levels

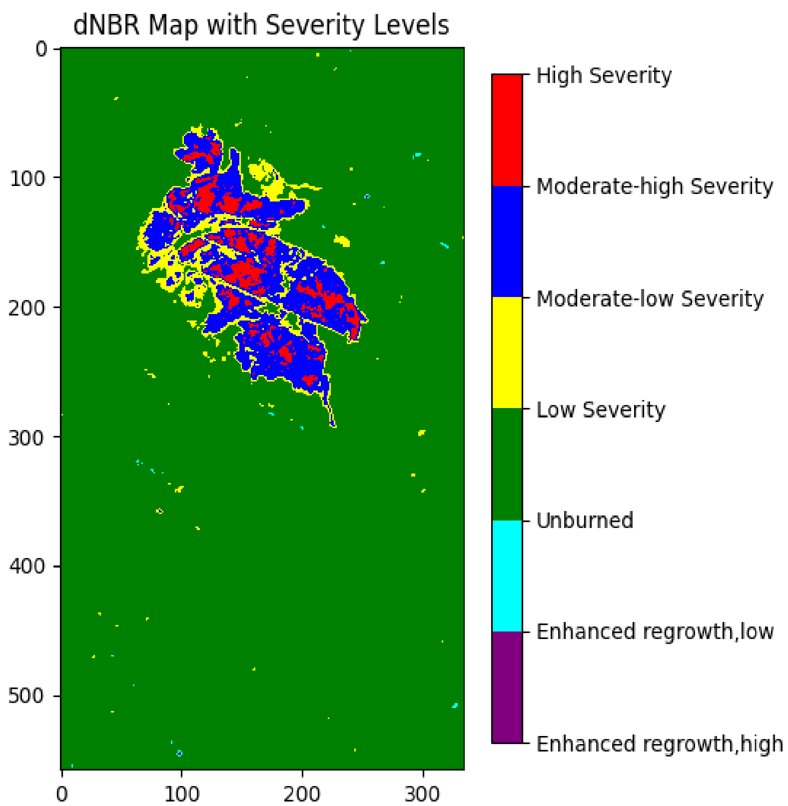


Figure 6. Differenced Normalised Burn Ratio map of Larache study area showing burn severity levels. Red represents areas of moderate-high burn severity to high severity, Blue represents areas of moderate-low to moderate-high burn severity, Yellow represents areas of low to moderate-low burn severity and green represents unburned areas.

From the data analysis, dNBR values ranged from -0.364 to 0.053 with an average value of 0.564(**Table 2**). The value range was then divided into seven classes following the USGS classification guidelines (**Table 1**). The prominent burn severity classes (**Figure 6**) in Larache were found to consist of moderate-high severity, moderate-low severity, low severity and unburned classes.

3.1.4. dNBR based Areas of Burned Classes

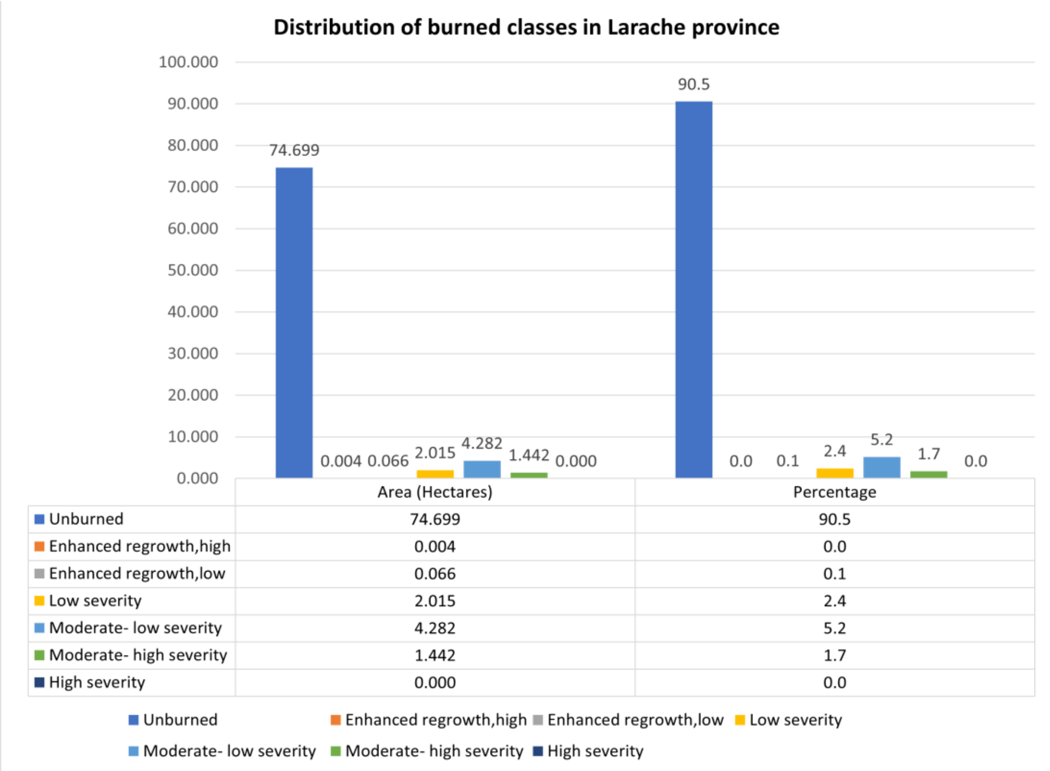


Figure 7. Distribution of Burn severity classes derived from dNBR index with their areas in hectares.

The results (**Figure 7**) indicated that 1.7% of the study area experienced moderate-high severity burns, 5.2% experienced moderate-low severity, 2.4% experienced low severity. 0.1% was experiencing low enhanced regrowth and 90.5% was unburned.

3.2. Normalised Differenced Vegetation Indices for Larache

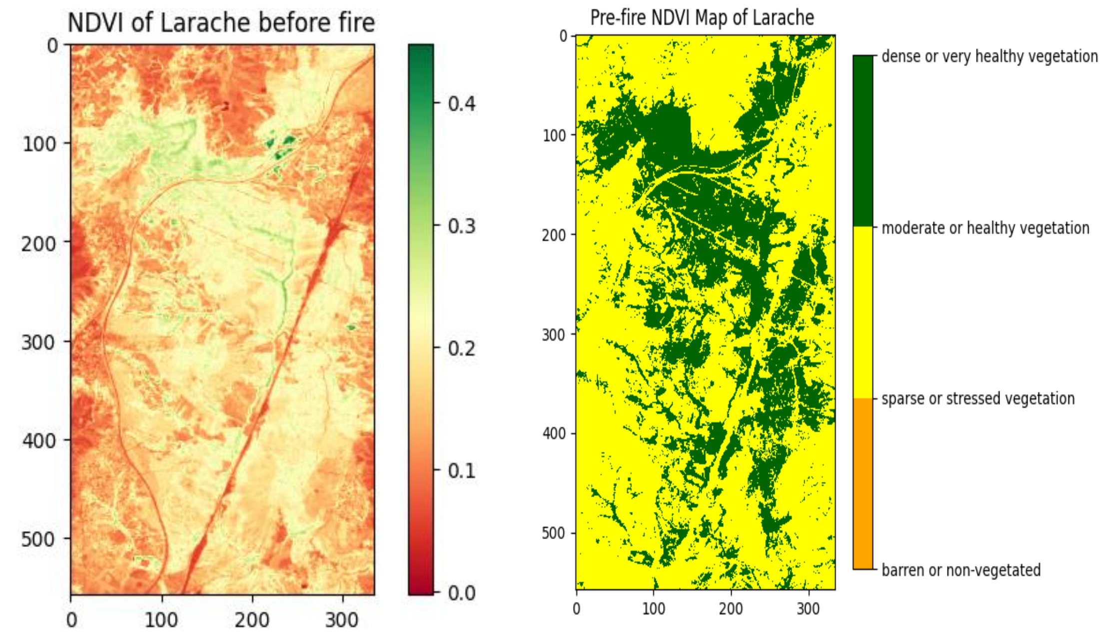
3.2.1.Summary of Normalised Differenced Vegetation Indices for Larache

	Pre-fire NDVI	Post-fire NDVI	dNDVI
Minimum	-0.003	-0.065	-0.310
Maximum	0.447	0.575	0.339
Mean	0.169	0.173	-0.003

Table 3. Pre-fire and post-fire Normalized Difference Vegetation Index values and differenced Normalised Burn Ratio values for Larache. The range of dNDVI values is (0.339+0.310) =0.649

From **Table 3**, the mean NDVI for the pre-fire image was 0.169 with a standard deviation of 0.054 while that of the post-fire image was 0.173 with a standard deviation of 0.08. The resulting mean dNDVI was -0.003 with a standard deviation of 0.065. This negative value of dNDVI shows there was a drop in vegetation cover over time caused by the burning wildfire.

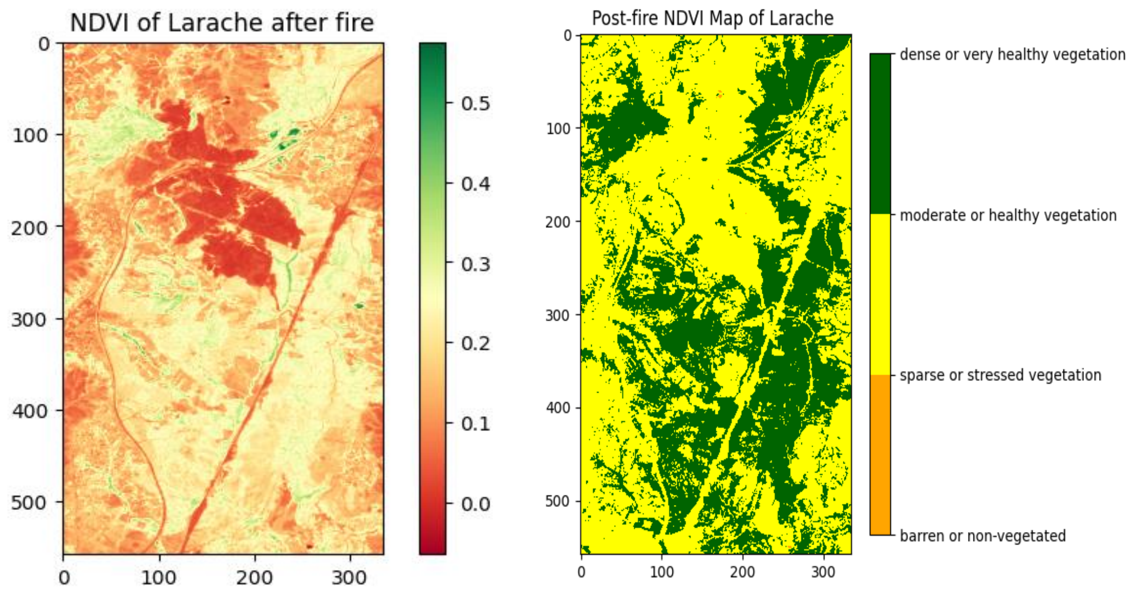
3.2.2. Mapping pre-fire NDVI of Larache



(**a**) (**b**)

Figure 8. (**a**) Normalised Difference Vegetation Index (NDVI) pre-fire map of Larache study area with Green representing areas with vegetation while Brown represents areas without vegetation; (**b**) Pre-fire NDVI classification map of Larache Green representing moderate or healthy vegetation to dense or very healthy vegetation, Yellow represents sparse or stressed vegetation and Orange representing barren or non-vegetated areas.

3.2.3. Mapping post- fire NDVI of Larache



(**a**) (**b**)

Figure 9. (**a**) Normalised Difference Vegetation Index (NDVI) post-fire map of Larache study area with Green representing areas with vegetation while Brown represents areas without vegetation;

(**b**) Post-fire NDVI classification map of Larache with Green representing moderate or healthy vegetation to dense or very healthy vegetation, Yellow represents sparse or stressed vegetation and Orange representing barren or non-vegetated areas.

3.2.4. Map of differenced Normalised Differenced Vegetation Index for Larache

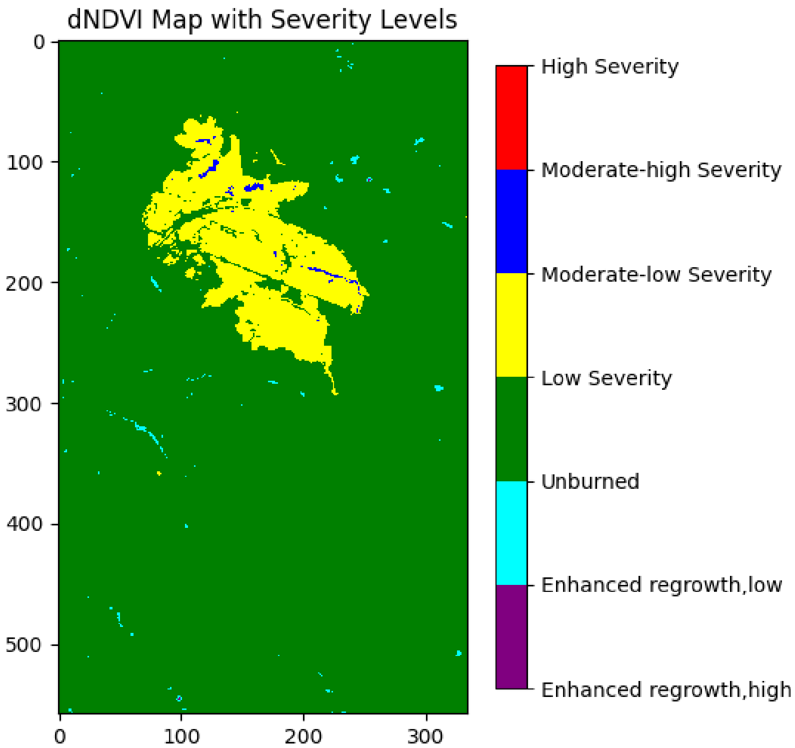
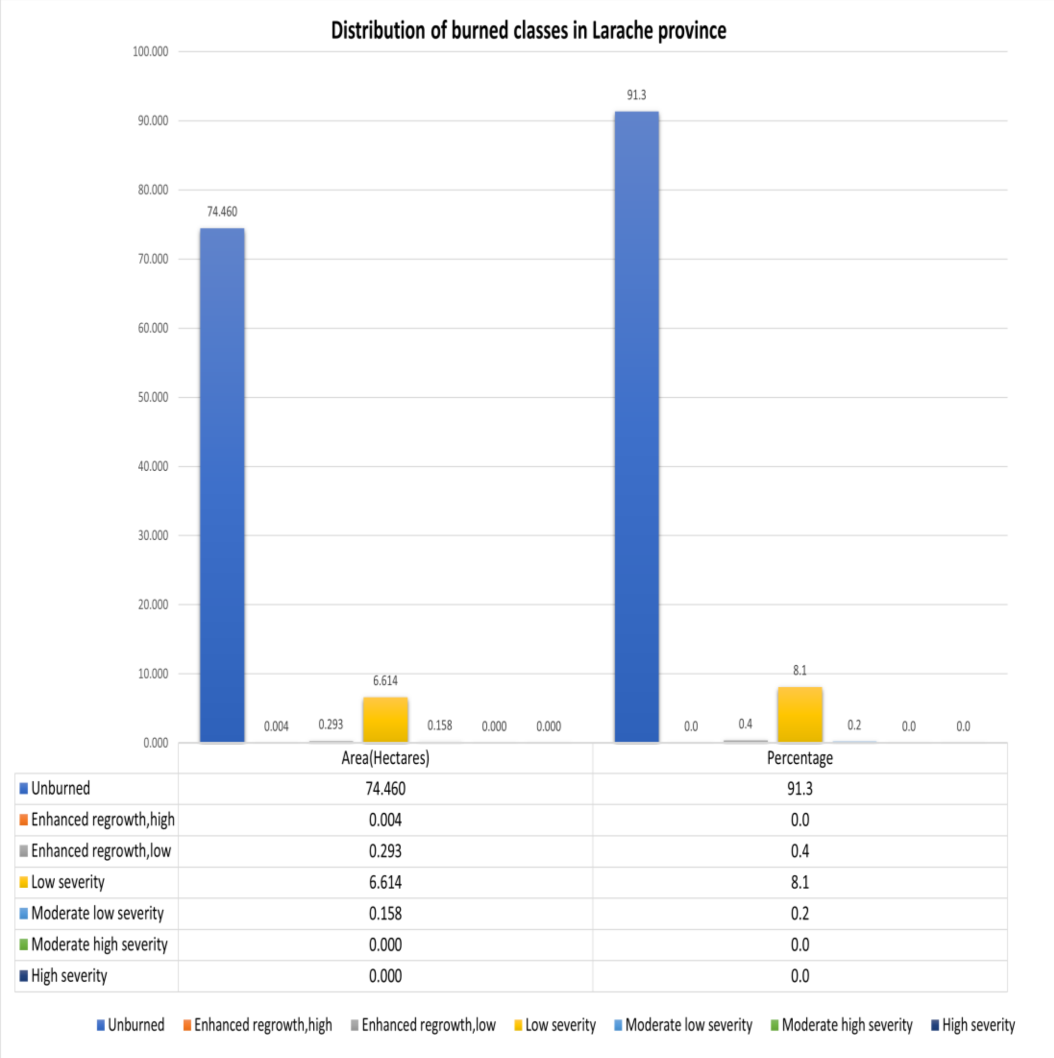


Figure 10. Differenced Normalised Difference Vegetation Index based map for Larache. Blue represents moderate-low burn severity areas, Yellow represents low-severity burn areas, Green represents unburned areas and cyan represents enhanced regrowth areas.

Comparing the areas in green representing moderate or healthy vegetation in the pre-fire NDVI map (**Figure 8)** with the similar areas in the post-fire NDVI map (**Figure 9**) reveals that a significant area of Northern Larache was burnt severely and was transformed into sparse or stressed vegetation by the wildfire. This change from green to yellow represents drop in NDVI or greenness of vegetation over time which was calculated for all pixels and mapped (**Figure 10**) to show the low severity burned scar in Yellow and the unburned areas in green (**Figure 10**).

3.2.5. dNDVI derived Areas of Burned Classes



	Area(Hectares)	Percentage
Unburned	74.460	91.3
Enhanced regrowth,high	0.004	0.0
Enhanced regrowth,low	0.293	0.4
Low severity	6.614	8.1
Moderate low severity	0.158	0.2
Moderate high severity	0.000	0.0
High severity	0.000	0.0

Figure 11. Distribution of Burn severity classes derived using dNDVI index with their areas in hectares.

The results in Figure 11 show that 91.3% of the study area was unburned, 8.1% experienced low severity burns and 0.2% experienced moderate low severity burns.

3.3 Relationship between dNDVI and dNBR

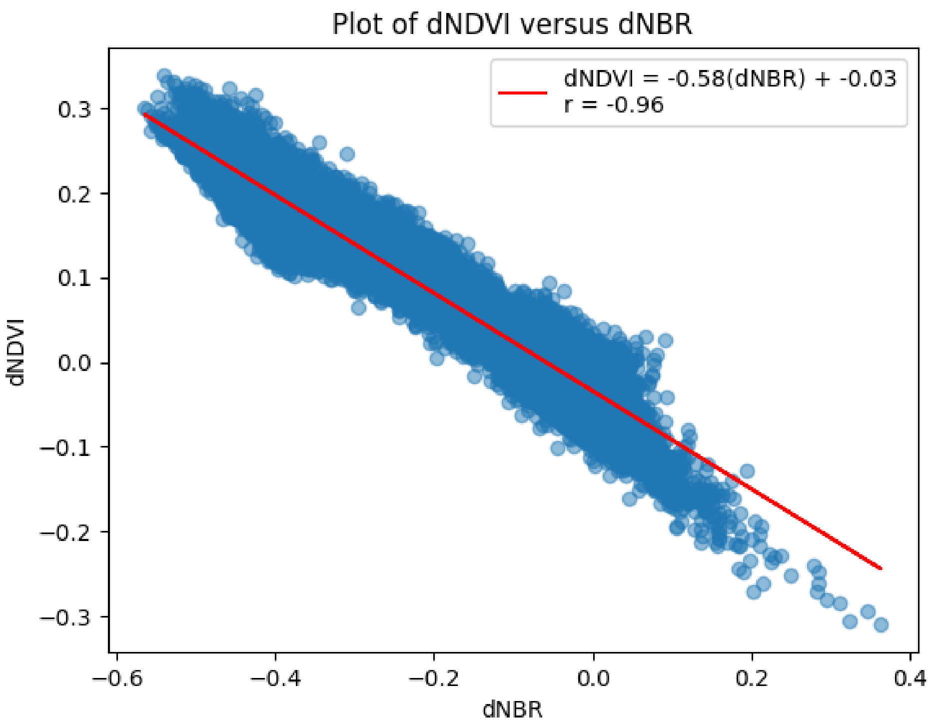


Figure 12. Plot of differenced Normalized Difference Vegetation Index against differenced Normalised Burn Ratio for Larache.

A scatter plot of dNDVI versus dNBR (**Figure 12**) was used to visualize the relationship between the recovery of greenness of vegetation after a fire and damage due to the fire. The plot has dNBR values ranging from -0.4 to +0.6 on the horizontal representing a progression from high enhanced regrowth and low enhanced regrowth after fire to unburned and high burn severity classes (**Table 1**). On the vertical axis, dNDVI values range from -0.3 to +0.3 representing a progression from decreased vegetation growth to increased vegetation growth over time. dNDVI and dNBR are inversely related so that as the severity of a burn (dNBR) increases, the health and density of vegetation (dNDVI) decreases. A very strong negative correlation of 0.96 exists between dNDVI and dNBR represented by a simple linear regression equation below.

$$\mathrm{dNDVI} = -0.58(\mathrm{dNBR}) - 0.03 \quad (5)$$

Figure 12. helps us understand how the recovery of vegetation after a wildfire is related to the burn severity and fire damage. The strong negative correlation between the two indices shows that the more severe the burn scar, the lesser the presence of healthy and green vegetation. In terms of vegetation recovery after a fire, high burn severity areas will have little vegetation regrowth while low burn severity areas experience high vegetation regrowth after a wildfire.

3.2. Average Normalised Burn Ratio for Polygons

Index	Burned	Unburned
Mean NBR1 (Pre-fire)	0.172	0.173
Mean NBR2 (Post-fire)	-0.139	0.172
Mean dNBR	0.310	0.001

Table 4. Pre-fire Normalised Burn Ratio, post-fire Normalised Burn Ratio and differenced Normalised Burn Ratio for burned and unburned polygons.

The burned polygon and unburned polygon were picked from the areas of burned forest and unburned forest (**Figure 2(b)** and **Figure 3(b)**) of the study area. The burned polygon had a mean dNBR value of 0.310 showing moderate-low burn severity according to the USGS severity classification (**Table 1**). The unburned polygon had a mean dNBR value of 0.001 representing the unburned class according to USGS (**Table 1**).

4. Discussion

The results of dNBR and dNDVI-derived maps (**Figure 6** and **Figure 10** respectively) show how accurate or sensitive each index is at detecting and classifying burn severity. dNDVI was able to detect and classify five classes namely; low and high enhanced regrowth, unburned, low severity, and moderate low severity areas (**Figure 11**) while dNBR detected and classified six classes (**Figure 7**); the moderate-high severity class in addition to the five detected by the dNDVI. This shows that dNBR is more sensitive and accurate in detecting burn severity than dNDVI. This fact is supported and agrees with previous studies [17,20] that support the superiority in accuracy of dNBR over RdNBR and dNDVI in burn severity assessment.

In agreement with [20], the results (**Figure 6** and **Figure 10**) show that both dNDVI and dNBR effectively discriminated between burned and unburned pixels. However, dNBR was better than dNDVI at separating the moderate burn severity class into low- and high-level severity [20]. In further agreement with [20], the post-fire NBR and post-fire NDVI (**Figure 5(b)** and **Figure 9(a)**) were effective in discriminating between high-severity pixels. The dNBR values had a range of 0.928, post-fire NBR values had a range of 0.0.868 (**Table 2**) and dNDVI values had a range of 0.649(**Table 3**). Given that dNBR and post-fire NBR indices presented higher ranges than dNDVI, the pair of them is more suitable for detecting varying levels of burn severity which is in agreement with [17,20].

5. Conclusions

dNDVI and dNBR are inversely related so that as the severity of a burn (dNBR) increases, the health and density of vegetation (dNDVI) decreases. The results show that dNBR is more accurate and robust than dNDVI at burn severity assessment in our study area. However, according to [17], these findings have shortcomings since the classification accuracies of each index was not calculated. Future areas of improvement in the methodology would include producer accuracy, user accuracy, overall accuracy and Kappa coefficients for accuracy quantification [17].

Funding: This research received no external funding

Data Availability Statement: Sentinel-2 data was downloaded and processed using Google Earth Engine available at https://earthengine.google.com/

Acknowledgments: I acknowledge the use of Google Earth Engine for data access, processing and download. I also acknowledge use of Google Colab for writing and running the Python code notebook.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Bowman, D. M. J. S.; Balch, J. K.; Artaxo, P.; Bond, W. J.; Cochrane, M. A.; D'Antonio, C. M.; DeFries, R.; Johnston, F. H.; Keeley, J. E.; Krawchuk, M. A.; et al. The Human Dimension of Fire Regimes on Earth. *J. Biogeogr*. **2011**, 38 (12), 2223–2236. DOI: 10.1111/j.1365-2699.2011.02595.x
2. Westerling, A. L. Increasing Western US Forest Wildfire Activity: Sensitivity to Changes in the Timing of Spring. *Philos. Trans. R. Soc. B Biol. Sci.* **2016**, 371 (1696), 20150178. DOI: 10.1098/rstb.2015.0178
3. Flannigan, M.; Cantin, A. S.; De Groot, W. J.; Wotton, M.; Newbery, A.; Gowman, L. M. Global Wildland Fire Season Severity in the 21st Century. *For. Ecol. Manag*. **2013**, 294, 54–61. DOI: 10.1016/j.foreco.2012.10.022
4. Lionello, P.; Scarascia, L. The Relation between Climate Change in the Mediterranean Region and Global Warming. *Reg. Environ. Change* **2018**, 18 (5), 1481–1493. DOI: 10.1007/s10113-018-1290-1
5. Chuvieco, E.; Aguado, I.; Yebra, M.; Nieto, H.; Salas, J.; Martín, M. P.; Vilar, L.; Martínez, J.; Martín, S.; Ibarra, P.; et al. Development of a Framework for Fire Risk Assessment Using Remote Sensing and Geographic Information System Technologies. *Ecol. Model*. **2010**, 221 (1), 46–58. DOI: 10.1016/j.ecolmodel.2008.11.017
6. Roy, D. P.; Boschetti, L.; Justice, C. O.; Ju, J. The Collection 5 MODIS Burned Area Product—Global Evaluation by Comparison with the MODIS Active Fire Product. *Remote Sens. Environ*. **2008**, 112 (9), 3690–3707. DOI: 10.1016/j.rse.2008.05.013
7. Pflugmacher, D.; Cohen, W. B.; Kennedy, R. E. Using Landsat-Derived Disturbance History (1972–2010) to Predict Current Forest Structure. *Remote Sens. Environ*. **2012**, 122, 146–165. DOI: 10.1016/j.rse.2011.09.025
8. Fuerst-Bjelis, B. The Fire in the Mediterranean Region: A Case Study of Forest Fires in Portugal. In Mediterranean Identities - *Environment, Society, Culture*; **2017**; Chapter 13. DOI: 10.5772/intechopen.69410
9. Key, C. H.; Benson, N. C. Landscape Assessment: Remote Sensing of Severity, the Normalized Burn Ratio. In FIREMON: Fire Effects Monitoring and Inventory System; USDA Forest Service, Rocky Mountain Research Station: Ogden, UT, **2006**; Vol. LA 1-LA 51.
10. Miller, J. D.; Thode, A. E. Quantifying Burn Severity in a Heterogeneous Landscape with a Relative Version of the Delta Normalized Burn Ratio (dNBR). *Remote Sens. Environ.* **2007**, 109 (1), 66–80. DOI: 10.1016/j.rse.2006.12.006
11. Drusch, M.; Del Bello, U.; Carlier, S.; Colin, O.; Fernandez, V.; Gascon, F.; Hoersch, B.; Isola, C.; Laberinti, P.; Martimort, P.; et al. Sentinel-2: ESA's Optical High-Resolution Mission for GMES Operational Services. *Remote Sens. Environ*. **2012**, 120, 25–36. DOI: 10.1016/j.rse.2011.11.026
12. Gorelick, N.; Hancher, M.; Dixon, M.; Ilyushchenko, S.; Thau, D.; Moore, R. Google Earth Engine: Planetary-Scale Geospatial Analysis for Everyone. *Remote Sens. Environ*. **2017**, 202, 18–27. DOI: 10.1016/j.rse.2017.06.031
13. Rouse, J.W.; Haas, R.H.; Schell, J.A.; Deering, D.W. Monitoring Vegetation Systems in the Great Plains with ERTS. In Proceedings of the Third Earth Resources Technology Satellite-1 Symposium, Washington, DC, USA, 10-14 December **1973**; Volume 351, pp. 309-317.
14. López-García, M. J.; Caselles, V. Mapping Burns and Natural Reforestation Using Thematic Mapper Data. *Geocarto Int.* **1991**, 6 (1), 31–37. DOI: 10.1080/10106049109354290
15. Epting, J.; Verbyla, D.; Sorbel, B. Evaluation of Remotely Sensed Indices for Assessing Burn Severity in Interior Alaska Using Landsat TM and ETM+. *Remote Sens. Environ*. **2005**, 96 (3-4), 328–339. DOI: 10.1016/j.rse.2005.03.002

16. Parks, S. A.; Dillon, G. K.; Miller, C. A New Metric for Quantifying Burn Severity: The Relativized Burn Ratio. *Remote Sens*. **2014**, 6 (3), 1827–1844. DOI: 10.3390/rs6031827

17. Soverel, N. O.; Perrakis, D. D.; Coops, N. C. Estimating Burn Severity from Landsat Differenced Normalized Burn Ratio (dNBR) and Normalized Difference Vegetation Index (NDVI) Data in Western Canada. *Remote Sens. Environ*. **2010**, 114 (8), 1656–1666. DOI: 10.1016/j.rse.2010.03.013

18. Bastarrika, A.; Chuvieco, E.; Martín, M. P. Mapping Burned Areas from Landsat TM/ETM+ Data with a Two-Date Algorithm: Balancing Commission and Omission Errors. *Remote Sens. Environ*. **2011**, 115 (4), 1003–1012. DOI: 10.1016/j.rse.2010.12.005

19. Llorens, R.; Sobrino, J. A.; Fernández, C.; Fernández-Alonso, J. M.; Vega, J. A. A Methodology to Estimate Forest Fires Burned Areas and Burn Severity Degrees Using Sentinel-2 Data. Application to the October 2017 Fires in the Iberian Peninsula. *Int. J. Appl. Earth Obs. Geoinf.* **2021**, 95, 102243. DOI: 10.1016/j.jag.2020.102243

20. Escuin, S.; Navarro, R.; Fernández, P. Fire Severity Assessment by Using NBR (Normalized Burn Ratio) and NDVI (Normalized Difference Vegetation Index) Derived from LANDSAT TM/ETM Images. Int. J. Remote Sens. **2008**, 29 (4), 1053-1073. DOI: 10.1080/01431160701281072

Appendix: Google Colab Python Notebook

Mapping fire damage from the wildfires in Larache Province, Morocco, in 2022 using Sentinel-2 Normalised Burn Ratio (NBR) and differenced NBR1

Connecting to Google Drive from Colab.

```
# Loading the Drive helper and mount your Google Drive as a drive in the virtual machine
from google.colab import drive
drive.mount('/content/drive')
Mounted at /content/drive
```

Importing all required libraries

```
# Installing some libraries that are not on Colab by default
!pip install rasterio
!pip install geopandas
!pip install rasterstats
!pip install earthengine-api
!pip install requests
!pip install sentinelsat

# Importing libraries
import geopandas as gpd
import rasterio
from rasterio import plot
from rasterio.plot import show_hist
import matplotlib.pyplot as plt
import numpy as np
from osgeo import gdal, ogr, osr
import json
import os
from os import listdir
from os.path import isfile, isdir, join
import math
from pprint import pprint
import shutil
import sys
import zipfile
import requests
```

```
import io
import webbrowser
import ee
import pandas as pd
import rasterio.mask
import xarray as xr
import matplotlib.colors as colors
import matplotlib
import glob
from rasterio.plot import show
from rasterio.mask import mask
from shapely.geometry import mapping
from scipy.stats import linregress

'''
------------------------------------------------------------
The following block of code originates and is modified from:

Balzter, H. (2023)
Materials for GY7709 masters computer classes.
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-
top%2FGY7709%5FSatellite%5FData%5FAnalysis%
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper-
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1
Downloaded 1 February 2023
------------------------------------------------------------
'''
# defining the root directory where our data are to be stored
rootdir = '/content/drive/MyDrive/satelliteCW1' # this is where pygge.py is saved on my
Google Drive

if rootdir not in sys.path:
    sys.path.append(rootdir)
# importing the pygge library of functions for this module
import pygge

%matplotlib inline
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/pub
lic/simple/
Collecting rasterio
```

```
  Downloading rasterio-1.3.6-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
 (20.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 20.0/20.0 MB 5
7.0 MB/s eta 0:00:00
Collecting affine (from rasterio)
  Downloading affine-2.4.0-py3-none-any.whl (15 kB)
Requirement already satisfied: attrs in /usr/local/lib/python3.10/dist-packages (from r
asterio) (23.1.0)
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from
 rasterio) (2022.12.7)
Requirement already satisfied: click>=4.0 in /usr/local/lib/python3.10/dist-packages (f
rom rasterio) (8.1.3)
Collecting cligj>=0.5 (from rasterio)
  Downloading cligj-0.7.2-py3-none-any.whl (7.1 kB)
Requirement already satisfied: numpy>=1.18 in /usr/local/lib/python3.10/dist-packages (
from rasterio) (1.22.4)
Collecting snuggs>=1.4.1 (from rasterio)
  Downloading snuggs-1.4.7-py3-none-any.whl (5.4 kB)
Collecting click-plugins (from rasterio)
  Downloading click_plugins-1.1.1-py2.py3-none-any.whl (7.5 kB)
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (f
rom rasterio) (67.7.2)
Requirement already satisfied: pyparsing>=2.1.6 in /usr/local/lib/python3.10/dist-packa
ges (from snuggs>=1.4.1->rasterio) (3.0.9)
Installing collected packages: snuggs, cligj, click-plugins, affine, rasterio
Successfully installed affine-2.4.0 click-plugins-1.1.1 cligj-0.7.2 rasterio-1.3.6 snug
gs-1.4.7
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/pub
lic/simple/
Collecting geopandas
  Downloading geopandas-0.13.0-py3-none-any.whl (1.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.1/1.1 MB 14.
6 MB/s eta 0:00:00
Collecting fiona>=1.8.19 (from geopandas)
  Downloading Fiona-1.9.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1
6.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 16.0/16.0 MB 9
.2 MB/s eta 0:00:00
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (fr
om geopandas) (23.1)
Requirement already satisfied: pandas>=1.1.0 in /usr/local/lib/python3.10/dist-packages
 (from geopandas) (1.5.3)
```

```
Collecting pyproj>=3.0.1 (from geopandas)
  Downloading pyproj-3.5.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (
7.7 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.7/7.7 MB 38.
8 MB/s eta 0:00:00
Requirement already satisfied: shapely>=1.7.1 in /usr/local/lib/python3.10/dist-package
s (from geopandas) (2.0.1)
Requirement already satisfied: attrs>=19.2.0 in /usr/local/lib/python3.10/dist-packages
 (from fiona>=1.8.19->geopandas) (23.1.0)
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from
 fiona>=1.8.19->geopandas) (2022.12.7)
Requirement already satisfied: click~=8.0 in /usr/local/lib/python3.10/dist-packages (f
rom fiona>=1.8.19->geopandas) (8.1.3)
Requirement already satisfied: click-plugins>=1.0 in /usr/local/lib/python3.10/dist-pac
kages (from fiona>=1.8.19->geopandas) (1.1.1)
Requirement already satisfied: cligj>=0.5 in /usr/local/lib/python3.10/dist-packages (f
rom fiona>=1.8.19->geopandas) (0.7.2)
Collecting munch>=2.3.2 (from fiona>=1.8.19->geopandas)
  Downloading munch-2.5.0-py2.py3-none-any.whl (10 kB)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist
-packages (from pandas>=1.1.0->geopandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages
(from pandas>=1.1.0->geopandas) (2022.7.1)
Requirement already satisfied: numpy>=1.21.0 in /usr/local/lib/python3.10/dist-packages
 (from pandas>=1.1.0->geopandas) (1.22.4)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from mun
ch>=2.3.2->fiona>=1.8.19->geopandas) (1.16.0)
Installing collected packages: pyproj, munch, fiona, geopandas
Successfully installed fiona-1.9.3 geopandas-0.13.0 munch-2.5.0 pyproj-3.5.0
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/pub
lic/simple/
Collecting rasterstats
  Downloading rasterstats-0.18.0-py3-none-any.whl (17 kB)
Requirement already satisfied: affine<3.0 in /usr/local/lib/python3.10/dist-packages (f
rom rasterstats) (2.4.0)
Requirement already satisfied: click>7.1 in /usr/local/lib/python3.10/dist-packages (fr
om rasterstats) (8.1.3)
Requirement already satisfied: cligj>=0.4 in /usr/local/lib/python3.10/dist-packages (f
rom rasterstats) (0.7.2)
Collecting fiona<1.9 (from rasterstats)
  Downloading Fiona-1.8.22-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (
16.6 MB)
```

```
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 16.6/16.6 MB 1
2.1 MB/s eta 0:00:00
Requirement already satisfied: numpy>=1.9 in /usr/local/lib/python3.10/dist-packages (f
rom rasterstats) (1.22.4)
Requirement already satisfied: rasterio>=1.0 in /usr/local/lib/python3.10/dist-packages
 (from rasterstats) (1.3.6)
Collecting simplejson (from rasterstats)
  Downloading simplejson-3.19.1-cp310-cp310-manylinux_2_5_x86_64.manylinux1_x86_64.many
linux_2_17_x86_64.manylinux2014_x86_64.whl (137 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 137.9/137.9 kB 15
.9 MB/s eta 0:00:00
Requirement already satisfied: shapely in /usr/local/lib/python3.10/dist-packages (from
 rasterstats) (2.0.1)
Requirement already satisfied: attrs>=17 in /usr/local/lib/python3.10/dist-packages (fr
om fiona<1.9->rasterstats) (23.1.0)
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from
 fiona<1.9->rasterstats) (2022.12.7)
Requirement already satisfied: click-plugins>=1.0 in /usr/local/lib/python3.10/dist-pac
kages (from fiona<1.9->rasterstats) (1.1.1)
Requirement already satisfied: six>=1.7 in /usr/local/lib/python3.10/dist-packages (fro
m fiona<1.9->rasterstats) (1.16.0)
Requirement already satisfied: munch in /usr/local/lib/python3.10/dist-packages (from f
iona<1.9->rasterstats) (2.5.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (f
rom fiona<1.9->rasterstats) (67.7.2)
Requirement already satisfied: snuggs>=1.4.1 in /usr/local/lib/python3.10/dist-packages
 (from rasterio>=1.0->rasterstats) (1.4.7)
Requirement already satisfied: pyparsing>=2.1.6 in /usr/local/lib/python3.10/dist-packa
ges (from snuggs>=1.4.1->rasterio>=1.0->rasterstats) (3.0.9)
Installing collected packages: simplejson, fiona, rasterstats
  Attempting uninstall: fiona
    Found existing installation: Fiona 1.9.3
    Uninstalling Fiona-1.9.3:
      Successfully uninstalled Fiona-1.9.3
Successfully installed fiona-1.8.22 rasterstats-0.18.0 simplejson-3.19.1
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/pub
lic/simple/
Requirement already satisfied: earthengine-api in /usr/local/lib/python3.10/dist-packag
es (0.1.350)
Requirement already satisfied: google-cloud-storage in /usr/local/lib/python3.10/dist-p
ackages (from earthengine-api) (2.8.0)
```

```
Requirement already satisfied: google-api-python-client>=1.12.1 in /usr/local/lib/python3.10/dist-packages (from earthengine-api) (2.84.0)
Requirement already satisfied: google-auth>=1.4.1 in /usr/local/lib/python3.10/dist-packages (from earthengine-api) (2.17.3)
Requirement already satisfied: google-auth-httplib2>=0.0.3 in /usr/local/lib/python3.10/dist-packages (from earthengine-api) (0.1.0)
Requirement already satisfied: httplib2<1dev,>=0.9.2 in /usr/local/lib/python3.10/dist-packages (from earthengine-api) (0.21.0)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from earthengine-api) (2.27.1)
Requirement already satisfied: google-api-core!=2.0.*,!=2.1.*,!=2.2.*,!=2.3.0,<3.0.0dev,>=1.31.5 in /usr/local/lib/python3.10/dist-packages (from google-api-python-client>=1.12.1->earthengine-api) (2.11.0)
Requirement already satisfied: uritemplate<5,>=3.0.1 in /usr/local/lib/python3.10/dist-packages (from google-api-python-client>=1.12.1->earthengine-api) (4.1.1)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from google-auth>=1.4.1->earthengine-api) (5.3.0)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.10/dist-packages (from google-auth>=1.4.1->earthengine-api) (0.3.0)
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.10/dist-packages (from google-auth>=1.4.1->earthengine-api) (1.16.0)
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.10/dist-packages (from google-auth>=1.4.1->earthengine-api) (4.9)
Requirement already satisfied: pyparsing!=3.0.0,!=3.0.1,!=3.0.2,!=3.0.3,<4,>=2.4.2 in /usr/local/lib/python3.10/dist-packages (from httplib2<1dev,>=0.9.2->earthengine-api) (3.0.9)
Requirement already satisfied: google-cloud-core<3.0dev,>=2.3.0 in /usr/local/lib/python3.10/dist-packages (from google-cloud-storage->earthengine-api) (2.3.2)
Requirement already satisfied: google-resumable-media>=2.3.2 in /usr/local/lib/python3.10/dist-packages (from google-cloud-storage->earthengine-api) (2.5.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->earthengine-api) (1.26.15)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->earthengine-api) (2022.12.7)
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.10/dist-packages (from requests->earthengine-api) (2.0.12)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->earthengine-api) (3.4)
Requirement already satisfied: googleapis-common-protos<2.0dev,>=1.56.2 in /usr/local/lib/python3.10/dist-packages (from google-api-core!=2.0.*,!=2.1.*,!=2.2.*,!=2.3.0,<3.0.0dev,>=1.31.5->google-api-python-client>=1.12.1->earthengine-api) (1.59.0)
```

```
Requirement already satisfied: protobuf!=3.20.0,!=3.20.1,!=4.21.0,!=4.21.1,!=4.21.2,!=4
.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.19.5 in /usr/local/lib/python3.10/dist-packages (
from google-api-core!=2.0.*,!=2.1.*,!=2.2.*,!=2.3.0,<3.0.0dev,>=1.31.5->google-api-pyth
on-client>=1.12.1->earthengine-api) (3.20.3)
Requirement already satisfied: google-crc32c<2.0dev,>=1.0 in /usr/local/lib/python3.10/
dist-packages (from google-resumable-media>=2.3.2->google-cloud-storage->earthengine-ap
i) (1.5.0)
Requirement already satisfied: pyasn1<0.6.0,>=0.4.6 in /usr/local/lib/python3.10/dist-p
ackages (from pyasn1-modules>=0.2.1->google-auth>=1.4.1->earthengine-api) (0.5.0)
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/pub
lic/simple/
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (2.2
7.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.10/dist-
packages (from requests) (1.26.15)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-pac
kages (from requests) (2022.12.7)
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.10/d
ist-packages (from requests) (2.0.12)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages
(from requests) (3.4)
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/pub
lic/simple/
Collecting sentinelsat
  Downloading sentinelsat-1.2.1-py3-none-any.whl (48 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 48.8/48.8 kB 3
.0 MB/s eta 0:00:00
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (fro
m sentinelsat) (2.27.1)
Requirement already satisfied: click>=7.1 in /usr/local/lib/python3.10/dist-packages (f
rom sentinelsat) (8.1.3)
Collecting html2text (from sentinelsat)
  Downloading html2text-2020.1.16-py3-none-any.whl (32 kB)
Collecting geojson>=2 (from sentinelsat)
  Downloading geojson-3.0.1-py3-none-any.whl (15 kB)
Requirement already satisfied: tqdm>=4.58 in /usr/local/lib/python3.10/dist-packages (f
rom sentinelsat) (4.65.0)
Collecting geomet (from sentinelsat)
  Downloading geomet-1.0.0-py3-none-any.whl (28 kB)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from geo
met->sentinelsat) (1.16.0)
```

```
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.10/dist-
packages (from requests->sentinelsat) (1.26.15)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-pac
kages (from requests->sentinelsat) (2022.12.7)
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.10/d
ist-packages (from requests->sentinelsat) (2.0.12)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages
(from requests->sentinelsat) (3.4)
Installing collected packages: html2text, geomet, geojson, sentinelsat
Successfully installed geojson-3.0.1 geomet-1.0.0 html2text-2020.1.16 sentinelsat-1.2.1
```

Setting up some directory paths on Google Drive

1. A shapefile of Larache province is in my Google Drive. I drew a polygon and saved it as a shapefile on http://www.geojson.io.

```
# Connecting to Google Earth Engine API
# This opens a web page where i  enter my account information and a verification code
is also provided. This code is what i paste into the terminal.
!earthengine authenticate

ee.Initialize()
To authorize access needed by Earth Engine, open the following URL in a web browser and
 follow the instructions. If the web browser does not start automatically, please manua
lly browse the URL below.

    https://code.earthengine.google.com/client-auth?scopes=https%3A//www.googleapis.com
/auth/earthengine%20https%3A//www.googleapis.com/auth/devstorage.full_control&request_i
d=twmhr4UDalOCouqwF2Y0MIQtfo9eyX1OZ2Dz3JAMYyA&tc=OoXrSQp4z0FUtOsxtSs8rJbXTb8wiRKhX5pYjv
FP6cc&cc=EuRBf2I1jjH17quMcxhdb_xqE6glxrecpELEwCgbrO4

The authorization workflow will generate a code, which you should paste in the box belo
w.
Enter verification code: 4/1AbUR2VP2WYIYV49woH8vir8w4hTYSfdMaTGVws3E8xj-66vhqTKzgik16gM

Successfully saved authorization token.

'''
--------------------------------------------------------------
The following block of code originates  from:
```

```
Balzter, H. (2023)
Materials for GY7709 masters computer classes.
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-
top%2FGY7709%5FSatellite%5FData%5FAnalysis%
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper-
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1
Downloaded 1 February 2023
--------------------------------------------------------------
'''

# set up your directories for the satellite data
# Note that we do all the downloading and data analysis on the temporary drive
#    on Colab. We will copy the output directory to our Google Drive at the end.
#    Colab has more disk space (about 40 GB free space) than Google Drive (15 GB).
#    However, the data on the Colab disk space are NOT kept when you log out.

# path to your Google Drive
print("Connected to data directory: " + rootdir)

# path to your temporary drive on the Colab Virtual Machine. This will be removed
#   when the session ends.
cd = "/content/work"

# directory for downloading the Sentinel-2 composites
# Note that we are using the 'join' function imported from the os library here
# It is an easy way of merging strings into a directory structure.
# It is clever and chooses the / or \ depending on whether you are on Windows or Linux.
downloaddir = join(cd, 'download') # where we save the downloaded images

# CAREFUL: This code removes the named directories and everything inside them to free
up space
# Because the downloaddir is in your temporary drive, it should be empty at the
#    start of the session.
# Note: shutil provides a lot of useful functions for file and directory management
try:
  shutil.rmtree(downloaddir)
except:
  print(downloaddir + " not found.")

# create the new directories, unless they already exist
```

```
os.makedirs(cd, exist_ok=True)
os.makedirs(downloaddir, exist_ok=True)

print("Connected to Colab temporary data directory: " + cd)

print("\nList of contents of " + rootdir)
for f in sorted(os.listdir(rootdir)):
  print(f)
Connected to data directory: /content/drive/MyDrive/satelliteCW1
/content/work/download not found.
Connected to Colab temporary data directory: /content/work

List of contents of /content/drive/MyDrive/satelliteCW1
.ipynb_checkpoints
229010645_GY7709_CW1 BACKUP2.ipynb
229010645_GY7709_CW1 BACKUP3.ipynb
229010645_GY7709_CW1.html
229010645_GY7709_CW1.ipynb
229010645_GY7709_CW1_backup.ipynb
229010645_GY7709_CW1_original.html
229010645_GY7709_CW2.ipynb
NBR_larache_after.tiff
NBR_larache_before.tiff
NDVI_larache_after.tiff
NDVI_larache_before.tiff
__pycache__
backup
burned_layers
dNBR.tif
dNBR.tiff
dNBR_warped.tif
larache_after
larache_before
larache_before_fire.tif
larache_before_fire_warped.tif
larache_new_shapefile
larache_shapefiles
leicestershire
oakham
practical_week32_SentinelSat.ipynb
pygge.ipynb
pygge.py
```

```
sencredentials.txt
taza
unburned_layers
```

Defining search parameters for the before fire image of Larache

2. modifying some of the parameters and uploading the shapefile of Larache called "POLYGON.shp" .

```
'''
--------------------------------------------------------------
The following block of code  is modified from:

Balzter, H. (2023)
Materials for GY7709 masters computer classes.
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-
top%2FGY7709%5FSatellite%5FData%5FAnalysis%
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper-
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1
Downloaded 1 February 2023
--------------------------------------------------------------
'''
# EDITING THE SEARCH OPTIONS BELOW

shapefile = join(rootdir, 'larache_new_shapefile', 'FULL_POLYGON.shp') # ESRI Shapefile
of the study area

# checking whether the shapefile exists
if os.path.exists(shapefile):
  print('Shapefile found: '+shapefile)
else:
  print('ERROR: Shapefile not found: '+shapefile)
  print('Upload a shapefile to your Google Drive directory: '+ rootdir)

# Defining a date range for the search
datefrom = '2022-07-01' # start date for imagery search
dateto   = '2022-07-10' # end date for imagery search
time_range = [datefrom, dateto] # format as a list

# Defining which cloud cover percentage we accept in the images
```

```
clouds = 8 # maximum acceptable cloud cover in
Shapefile found: /content/drive/MyDrive/satelliteCW1/larache_new_shapefile/FULL_POLYGON
.shp
```

Getting some information about the shapefile.

```

'''
---------------------------------------------------------------
The following block of code originates from:

Balzter, H. (2023)
Materials for GY7709 masters computer classes.
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-
top%2FGY7709%5FSatellite%5FData%5FAnalysis%
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper-
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1
Downloaded 1 February 2023
---------------------------------------------------------------
'''
# Getting the shapefile layer's extent, CRS and EPSG code
extent, outSpatialRef, epsg = pygge.get_shp_extent(shapefile)
print("Extent of the area of interest (shapefile):\n", extent)
print(type(extent))
print("\nCoordinate referencing system (CRS) of the shapefile:\n", outSpatialRef)
print('EPSG code: ', epsg)
Extent of the area of interest (shapefile):
 (-6.08, -6.02, 35.21, 35.31)
<class 'tuple'>

Coordinate referencing system (CRS) of the shapefile:
 GEOGCS["WGS 84",
    DATUM["WGS_1984",
        SPHEROID["WGS 84",6378137,298.257223563,
            AUTHORITY["EPSG","7030"]],
        AUTHORITY["EPSG","6326"]],
    PRIMEM["Greenwich",0,
        AUTHORITY["EPSG","8901"]],
    UNIT["degree",0.0174532925199433,
        AUTHORITY["EPSG","9122"]],
```

```
    AXIS["Latitude",NORTH],
    AXIS["Longitude",EAST],
    AUTHORITY["EPSG","4326"]]
EPSG code:  4326
```

3. Getting the extent of the shapefile into a format that Google Earth Engine understands.

```

'''
--------------------------------------------------------------
The following block of code originates  from:

Balzter, H. (2023)
Materials for GY7709 masters computer classes.
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-
top%2FGY7709%5FSatellite%5FData%5FAnalysis%
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper-
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1
Downloaded 1 February 2023
--------------------------------------------------------------
'''
# GEE needs a special format for defining an area of interest.
# It has to be a GeoJSON Polygon and the coordinates should be first defined in a list
and then converted using ee.Geometry.
extent_list = list(extent)
print(extent_list)
print(type(extent_list))
# close the list of polygon coordinates by adding the starting node at the end again
# and make list elements in the form of coordinate pairs (y,x)
area_list = list([(extent[0], extent[2]),(extent[1], extent[2]),(extent[1], ex-
tent[3]),(extent[0], extent[3]),(extent[0], extent[2])])
print(area_list)
print(type(area_list))

search_area = ee.Geometry.Polygon(area_list)
print(search_area)
print(type(search_area))
[-6.08, -6.02, 35.21, 35.31]
<class 'list'>
[(-6.08, 35.21), (-6.02, 35.21), (-6.02, 35.31), (-6.08, 35.31), (-6.08, 35.21)]
<class 'list'>
ee.Geometry({
```

```
  "functionInvocationValue": {
    "functionName": "GeometryConstructors.Polygon",
    "arguments": {
      "coordinates": {
        "constantValue": [
          [
            [
              -6.08,
              35.21
            ],
            [
              -6.02,
              35.21
            ],
            [
              -6.02,
              35.31
            ],
            [
              -6.08,
              35.31
            ],
            [
              -6.08,
              35.21
            ]
          ]
        ]
      },
      "evenOdd": {
        "constantValue": true
      }
    }
  }
})
<class 'ee.geometry.Geometry'>
```

4. Accessing the Sentinel-2 collection on Google Earth Engine and running our search.

```

'''
---------------------------------------------------------------
The following block of code originates and is modified from:
```

```
Balzter, H. (2023)
Materials for GY7709 masters computer classes.
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-
top%2FGY7709%5FSatellite%5FData%5FAnalysis%
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper-
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1
Downloaded 1 February 2023
--------------------------------------------------------------
'''
# Obtaining download links for image composites from an image collection on Google
Earth Engine


# Name of the Sentinel 2 image collection
s2collection = ('COPERNICUS/S2')

# getting the median composite of Sentinel-2 images in the time range
s2median = pygge.obtain_image_sentinel(s2collection, time_range, search_area, clouds)

# Downloading the followig bands as a list of strings namely; Blue, Green,Red,NIR and
SWIR

bands = [ 'B2','B3','B4', 'B8','B12']
print(bands)

# spatial resolution of the downloaded data
resolution = 20 # in units of metres

# Download images in Geotiff, using the get_url(name, image, scale, region) method
# 'region' is obtained from the area, but the format is adjusted using get_region(geom)
method
search_region = pygge.get_region(search_area)
s2url = pygge.get_url('s2', s2median.select(bands), resolution, search_region, filePer-
Band=False)
print(s2url)
['B2', 'B3', 'B4', 'B8', 'B12']
https://earthengine.googleapis.com/v1alpha/projects/earthengine-legacy/thumbnails/44e3d
97d9adfca34655516a4c2fea849-11e933b4d0f41ebc8d8b877d0589bca7:getPixels
```

Download the data for the before fire image of Larache

5. The next cell downloads the image composite as a zip file and unzips it.

```
'''
--------------------------------------------------------------
The following block of code originates  from:

Balzter, H. (2023)
Materials for GY7709 masters computer classes.
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-
top%2FGY7709%5FSatellite%5FData%5FAnalysis%
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper-
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1
Downloaded 1 February 2023
--------------------------------------------------------------
'''
# change directory to download directory
os.chdir(downloaddir)

# requesting information on the file to be downloaded
f = pygge.requests.get(s2url, stream =True)

# checking whether it is a zip file
check = zipfile.is_zipfile(io.BytesIO(f.content))

# either download the file as is, or unzip it
while not check:
    f = requests.get(s2url, stream =True)
    check = zipfile.is_zipfile(io.BytesIO(f.content))
else:
    z = zipfile.ZipFile(io.BytesIO(f.content))
    z.extractall()
```

Exploring the data directory structure of the downloaded files

```
'''
--------------------------------------------------------------
The following block of code originates and is modified from:

```

```
Balzter, H. (2023)
Materials for GY7709 masters computer classes.
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-
top%2FGY7709%5FSatellite%5FData%5FAnalysis%
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper-
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1
Downloaded 1 February 2023
--------------------------------------------------------------
'''
# where  the downloaded Sentinel-2 images are stored
os.chdir(downloaddir)
print("contents of ", downloaddir, ":")
!ls -l
contents of  /content/work/download :
total 2120
-rw-r--r-- 1 root root 2168674 May 12 13:28 s2.tif
```

6. The downloaded file " s2.tif " is seen.

7. the downloaded images are saved to a temporary directory that will be deleted when the virtual machine is closed. To save the images to my local directory, this is how it went;

8. I went to my Google Colab folder in the panel on the left hand side.

9. Found the download directory and clicked on a Sentinel-2 image folder.

10. Right-clicked on it ,renamed it and selected 'download' to save it.

Showing the before fire image of Larache as a colour composite

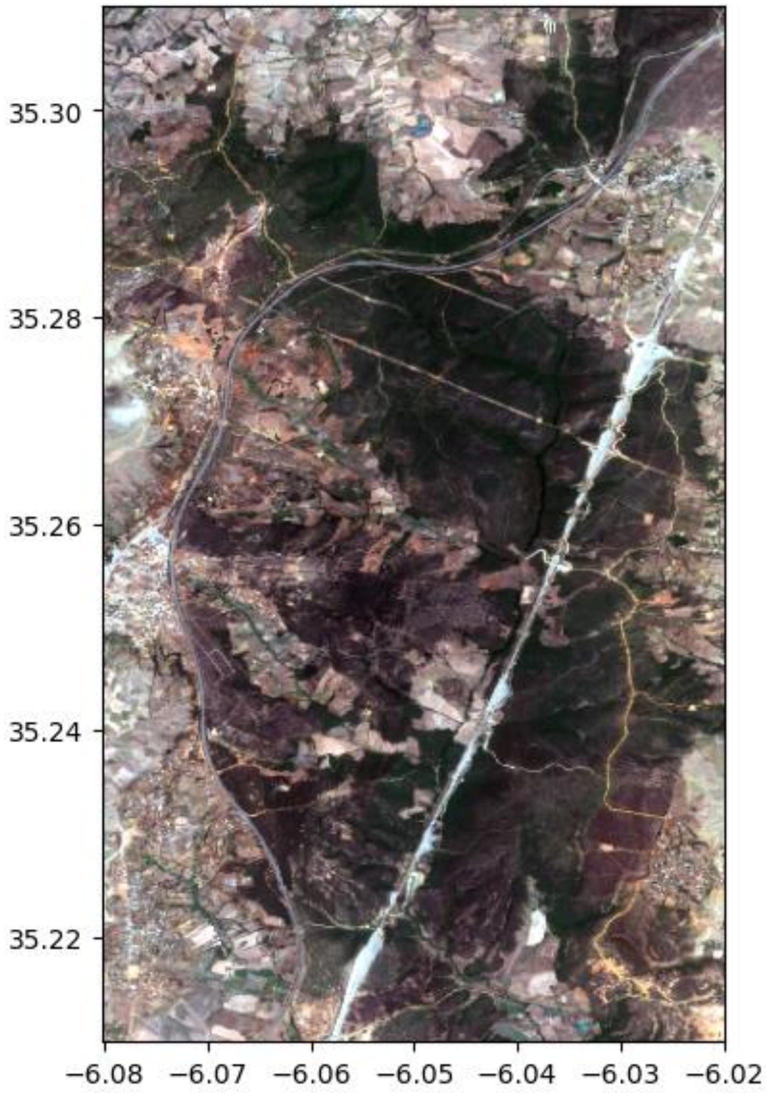
```
'''
--------------------------------------------------------------
The following block of code originates and is modified from:

Balzter, H. (2023)
Materials for GY7709 masters computer classes.
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-
top%2FGY7709%5FSatellite%5FData%5FAnalysis%
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper-
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%
```

```
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1
Downloaded 1 February 2023
--------------------------------------------------------------
'''
# getting list of all tiff files in the directory
allfiles = [f for f in listdir(downloaddir) if isfile(join(downloaddir, f))]
print(allfiles)

# selecting the file for visualisation
thisfile = allfiles[0]
print(thisfile)
['s2.tif']
s2.tif
```

True Colour Composite of before fire image of Larache

```
'''
--------------------------------------------------------------
The following block of code originates and is modified from:

Balzter, H. (2023)
Materials for GY7709 masters computer classes.
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-
top%2FGY7709%5FSatellite%5FData%5FAnalysis%
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper-
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1
Downloaded 1 February 2023
--------------------------------------------------------------
'''
# creating a figure
fig, (ax1) = plt.subplots(1,1, figsize=(7,7))
fig.patch.set_facecolor('white')

# the downloaded file is float32 data format
# for plotting,  uint8 data format is needed

# plotting the image with full extent
pygge.easy_plot(thisfile, ax=ax1, bands=[3,2,1], percentiles=[0,99])
```

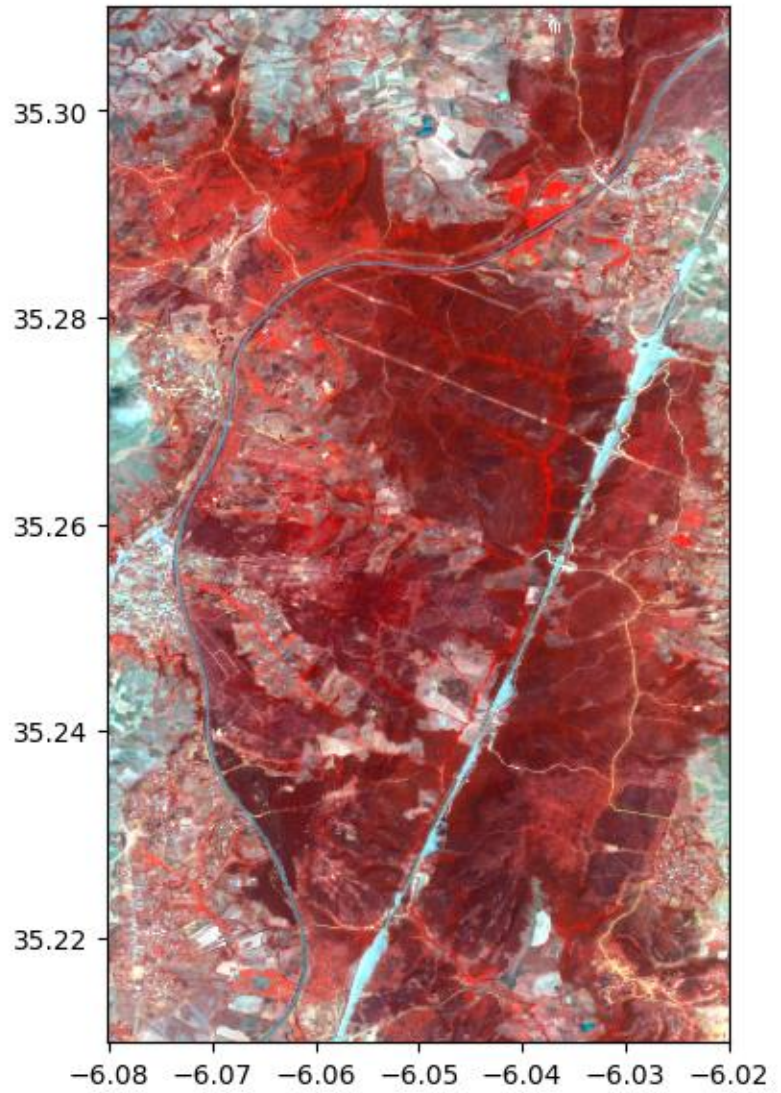
```
WARNING:rasterio._env:CPLE_AppDefined in s2.tif: TIFFReadDirectory:Sum of Photometric t
ype-related color channels and ExtraSamples doesn't match SamplesPerPixel. Defining non
-color channels as ExtraSamples.
WARNING:rasterio._env:CPLE_AppDefined in TIFFReadDirectory:Sum of Photometric type-rela
ted color channels and ExtraSamples doesn't match SamplesPerPixel. Defining non-color c
hannels as ExtraSamples.
```



False Colour Composite of before fire image of Larache

```
'''
--------------------------------------------------------------
The following block of code originates and is modified from:

Balzter, H. (2023)
Materials for GY7709 masters computer classes.
```

```
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-
top%2FGY7709%5FSatellite%5FData%5FAnalysis%
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper-
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1
Downloaded 1 February 2023
--------------------------------------------------------------
'''
# creating a figure with 2x3 subplots
fig, (ax1) = plt.subplots(1,1, figsize=(7,7))
fig.patch.set_facecolor('white')

# the downloaded file is float32 data format
# for plotting,  uint8 data format is needed

# plotting the image with full extent
pygge.easy_plot(thisfile, ax=ax1, bands=[4,2,1], percentiles=[0,99])
WARNING:rasterio._env:CPLE_AppDefined in s2.tif: TIFFReadDirectory:Sum of Photometric t
ype-related color channels and ExtraSamples doesn't match SamplesPerPixel. Defining non
-color channels as ExtraSamples.
WARNING:rasterio._env:CPLE_AppDefined in TIFFReadDirectory:Sum of Photometric type-rela
ted color channels and ExtraSamples doesn't match SamplesPerPixel. Defining non-color c
hannels as ExtraSamples.
```



Warping the downloaded before fire image composite of Larache into another map projection

11. The coordinate reference system (CRS) of the downloaded image composite is not in the UK national map projection. hence it is reprojected.

```
'''
--------------------------------------------------------------
The following block of code originates and is modified from:

Balzter, H. (2023)
Materials for GY7709 masters computer classes.
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15
```

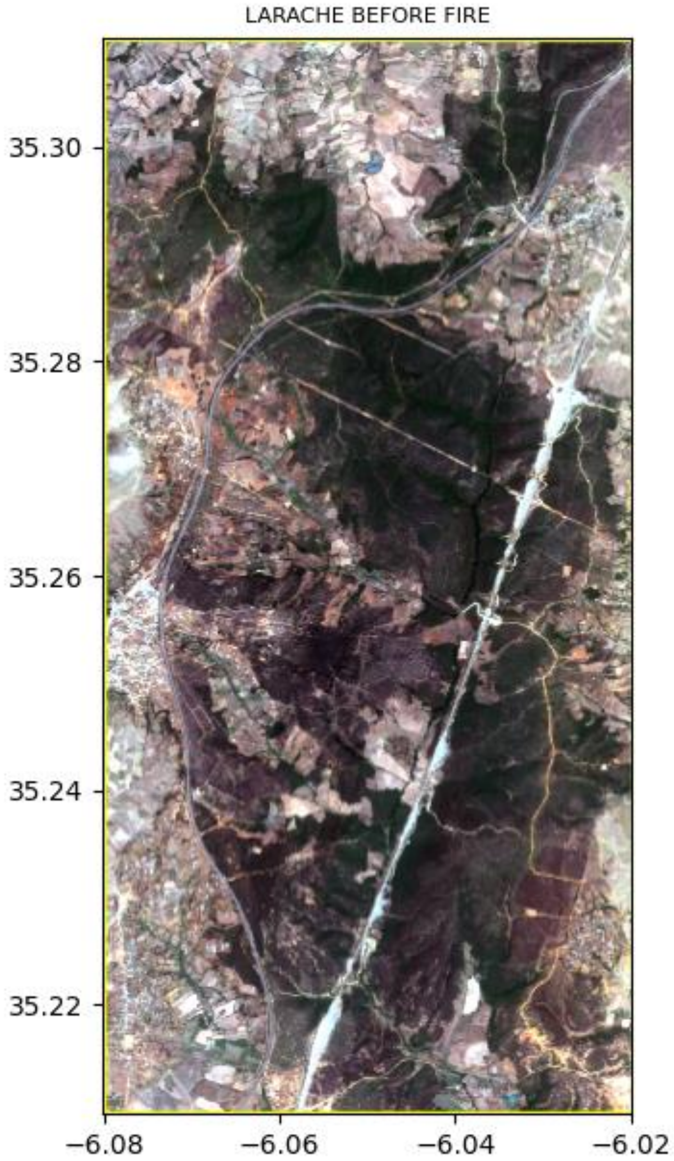
```
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-
top%2FGY7709%5FSatellite%5FData%5FAnalysis%
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper-
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1
Downloaded 1 February 2023
--------------------------------------------------------------
'''
# printing the EPSG code of our shapefile into which we want to reproject the TCI im-
ages
print("Reprojecting image to EPSG projection ", epsg)

# making a file name for the new file
warpfile = thisfile.split(sep='.')[0] + '_warped.tif'
print("We are in this directory: ")
!pwd
print("Input file: ", thisfile)
print("Output file: ", warpfile)

# calling the easy_warp function
tmp = pygge.easy_warp(thisfile, warpfile, epsg)
Reprojecting image to EPSG projection  4326
We are in this directory:
/content/work/download
WARNING:rasterio._env:CPLE_AppDefined in s2.tif: TIFFReadDirectory:Sum of Photometric t
ype-related color channels and ExtraSamples doesn't match SamplesPerPixel. Defining non
-color channels as ExtraSamples.
WARNING:rasterio._env:CPLE_AppDefined in TIFFReadDirectory:Sum of Photometric type-rela
ted color channels and ExtraSamples doesn't match SamplesPerPixel. Defining non-color c
hannels as ExtraSamples.
Input file:  s2.tif
Output file:  s2_warped.tif
Creating warped file:s2_warped.tif
```

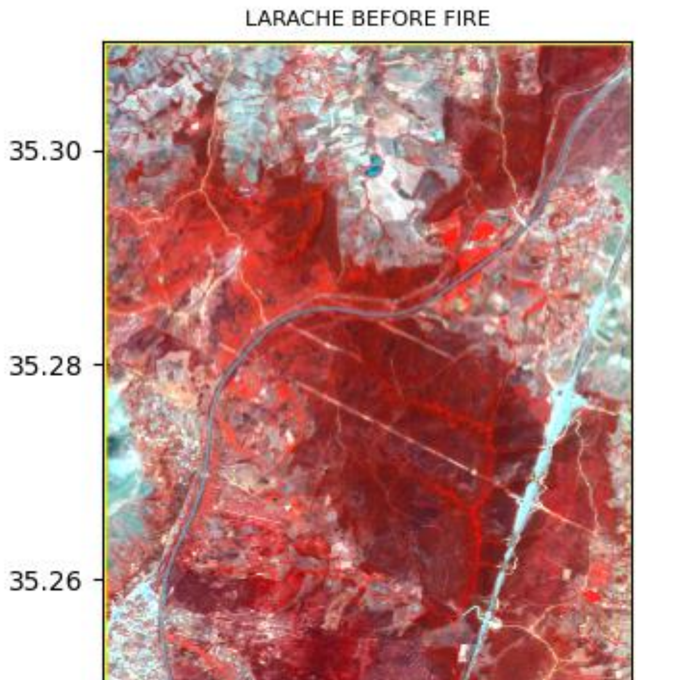
Plotting the shapefile on top of the raster

12. To see the locations of our polygons on top of our image composite, we can do that with the Geopandas library.

```
'''
--------------------------------------------------------------
The following block of code originates and is modified from:
```

```
Balzter, H. (2023)
Materials for GY7709 masters computer classes.
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-
top%2FGY7709%5FSatellite%5FData%5FAnalysis%
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper-
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1
Downloaded 1 February 2023
--------------------------------------------------------------
'''
# creating a figure with subplots
fig, ax = plt.subplots(2,1, figsize=(10,16))
fig.patch.set_facecolor('white')

# plotting the image with full extent in true colour
pygge.easy_plot(warpfile, ax=ax[0], percentiles=[0,98], bands=[3,2,1],
                shapefile=shapefile, fillcolor="none", linecolor="yellow",
                title="LARACHE BEFORE FIRE")

# plotting the image with full extent in false colour
pygge.easy_plot(warpfile, ax=ax[1], percentiles=[0,98], bands=[4,2,1],
                shapefile=shapefile, fillcolor="none", linecolor="yellow",
                title="LARACHE BEFORE FIRE")
```


LARACHE BEFORE FIRE
35.30
35.28
35.26
35.24
35.22
−6.08
−6.06
−6.04
−6.02


LARACHE BEFORE FIRE
35.30
35.28
35.26

Defining the search parameters for the after fire image of Larache

13. We change the search dates to get a new image after the fire had burnt out. Fires were first reported on 15 July 2022

```
'''
--------------------------------------------------------------
The following block of code originates and is modified from:

Balzter, H. (2023)
Materials for GY7709 masters computer classes.
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-
top%2FGY7709%5FSatellite%5FData%5FAnalysis%
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper-
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1
Downloaded 1 February 2023
--------------------------------------------------------------
'''
# EDITing THE SEARCH OPTIONS BELOW

shapefile = join(rootdir, 'larache_shapefiles', 'FULL_POLYGON.shp') # ESRI Shapefile of 
the study area

# checking whether the shapefile exists
if os.path.exists(shapefile):
  print('Shapefile found: '+shapefile)
else:
  print('ERROR: Shapefile not found: '+shapefile)
  print('Upload a shapefile to your Google Drive directory: '+ rootdir)

# Defining a date range for the search
datefrom = '2022-07-30' # start date for imagery search
dateto   = '2022-08-05' # end date for imagery search
time_range = [datefrom, dateto] # format as a list

# Defining percentage cloud cover accepted in the images
clouds = 10 # maximum acceptable cloud cover in %
Shapefile found: /content/drive/MyDrive/satelliteCW1/larache_shapefiles/FULL_POLYGON.sh
p
```

14. Get some information about our shapefile.

```
'''
--------------------------------------------------------------
The following block of code originates and is modified from:

Balzter, H. (2023)
Materials for GY7709 masters computer classes.
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-
top%2FGY7709%5FSatellite%5FData%5FAnalysis%
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper-
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1
Downloaded 1 February 2023
--------------------------------------------------------------
'''
# Getting the shapefile layer's extent, CRS and EPSG code
extent, outSpatialRef, epsg = pygge.get_shp_extent(shapefile)
print("Extent of the area of interest (shapefile):\n", extent)
print(type(extent))
print("\nCoordinate referencing system (CRS) of the shapefile:\n", outSpatialRef)
print('EPSG code: ', epsg)
Extent of the area of interest (shapefile):
 (-6.08, -6.02, 35.21, 35.31)
<class 'tuple'>

Coordinate referencing system (CRS) of the shapefile:
 GEOGCS["WGS 84",
    DATUM["WGS_1984",
        SPHEROID["WGS 84",6378137,298.257223563,
            AUTHORITY["EPSG","7030"]],
        AUTHORITY["EPSG","6326"]],
    PRIMEM["Greenwich",0,
        AUTHORITY["EPSG","8901"]],
    UNIT["degree",0.0174532925199433,
        AUTHORITY["EPSG","9122"]],
    AXIS["Latitude",NORTH],
    AXIS["Longitude",EAST],
    AUTHORITY["EPSG","4326"]]
EPSG code:  4326
```

15. Getting the extent of the shapefile into a format that Google Earth Engine understands.

```
'''
--------------------------------------------------------------
The following block of code originates and is modified from:

Balzter, H. (2023)
Materials for GY7709 masters computer classes.
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-
top%2FGY7709%5FSatellite%5FData%5FAnalysis%
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper-
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1
Downloaded 1 February 2023
--------------------------------------------------------------
'''
# GEE needs a special format for defining an area of interest.
# It has to be a GeoJSON Polygon and the coordinates should be first defined in a list
and then converted using ee.Geometry.
extent_list = list(extent)
print(extent_list)
print(type(extent_list))
# close the list of polygon coordinates by adding the starting node at the end again
# and make list elements in the form of coordinate pairs (y,x)
area_list = list([(extent[0], extent[2]),(extent[1], extent[2]),(extent[1], ex-
tent[3]),(extent[0], extent[3]),(extent[0], extent[2])])
print(area_list)
print(type(area_list))

search_area = ee.Geometry.Polygon(area_list)
print(search_area)
print(type(search_area))
[-6.08, -6.02, 35.21, 35.31]
<class 'list'>
[(-6.08, 35.21), (-6.02, 35.21), (-6.02, 35.31), (-6.08, 35.31), (-6.08, 35.21)]
<class 'list'>
ee.Geometry({
  "functionInvocationValue": {
    "functionName": "GeometryConstructors.Polygon",
    "arguments": {
      "coordinates": {
        "constantValue": [
```

```
          [
            [
              -6.08,
              35.21
            ],
            [
              -6.02,
              35.21
            ],
            [
              -6.02,
              35.31
            ],
            [
              -6.08,
              35.31
            ],
            [
              -6.08,
              35.21
            ]
          ]
        ]
      },
      "evenOdd": {
        "constantValue": true
      }
    }
  }
})
<class 'ee.geometry.Geometry'>
```

16. **Accessing the Sentinel-2 collection on Google Earth Engine and running the search.**

```
'''
--------------------------------------------------------------
The following block of code originates and is modified from:

Balzter, H. (2023)
Materials for GY7709 masters computer classes.
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-
top%2FGY7709%5FSatellite%5FData%5FAnalysis%
```

```
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper-
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1
Downloaded 1 February 2023
--------------------------------------------------------------
'''
# Obtain download links for image composites from an image collection on Google Earth
Engine

# Name of the Sentinel 2 image collection
s2collection = ('COPERNICUS/S2')

# getting the median composite of Sentinel-2 images in the time range
s2median = pygge.obtain_image_sentinel(s2collection, time_range, search_area, clouds)

# Blue, Green,Red,NIR and SWIR are downloaded
bands = ['B2','B3','B4', 'B8','B12' ]
print(bands)

# spatial resolution of the downloaded data
resolution = 20 # in units of metres

# Downloading images in Geotiff, using the get_url(name, image, scale, region) method
# 'region' is obtained from the area, but the format has to be adjusted using get_re-
gion(geom) method
search_region = pygge.get_region(search_area)
s2url = pygge.get_url('s2', s2median.select(bands), resolution, search_region, filePer-
Band=False)
print(s2url)
['B2', 'B3', 'B4', 'B8', 'B12']
https://earthengine.googleapis.com/v1alpha/projects/earthengine-legacy/thumbnails/3b62a
644518dacef2cf989403d8c3e3f-44432a28d67b784f448895200db8318e:getPixels
```

Downloading the data

17. The next cell downloads the image composite as a zip file and unzips it.

```
'''
--------------------------------------------------------------
The following block of code originates and is modified from:
```

```
Balzter, H. (2023)
Materials for GY7709 masters computer classes.
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-
top%2FGY7709%5FSatellite%5FData%5FAnalysis%
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper-
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1
Downloaded 1 February 2023
---------------------------------------------------------------
'''
# changing directory to download directory
os.chdir(downloaddir)

# requesting information on the file to be downloaded
f = pygge.requests.get(s2url, stream =True)

# checking whether it is a zip file
check = zipfile.is_zipfile(io.BytesIO(f.content))

# either download the file as is, or unzip it
while not check:
    f = requests.get(s2url, stream =True)
    check = zipfile.is_zipfile(io.BytesIO(f.content))
else:
    z = zipfile.ZipFile(io.BytesIO(f.content))
    z.extractall()
```

Exploring the data directory structure of our downloaded files

```
'''
---------------------------------------------------------------
The following block of code originates and is modified from:

Balzter, H. (2023)
Materials for GY7709 masters computer classes.
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-
top%2FGY7709%5FSatellite%5FData%5FAnalysis%
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper-
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%
```

```
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1
Downloaded 1 February 2023
----------------------------------------------------------------
'''
# where we stored the downloaded Sentinel-2 images
os.chdir(downloaddir)
print("contents of ", downloaddir, ":")
!ls -l
contents of  /content/work/download :
total 2200
-rw-r--r-- 1 root root 2251399 Apr 27 11:10 s2.tif
```

18. the downloaded file is " s2.tif ".

19. the downloaded images are saved to a temporary directory that will be deleted when the virtual machine is closed. To save the images to my local directory, this is how it went;

20. I went to my Google Colab folder in the panel on the left hand side.

21. Found the download directory and clicked on a Sentinel-2 image folder.

22. Right-clicked on it ,renamed it and selected 'download' to save it.

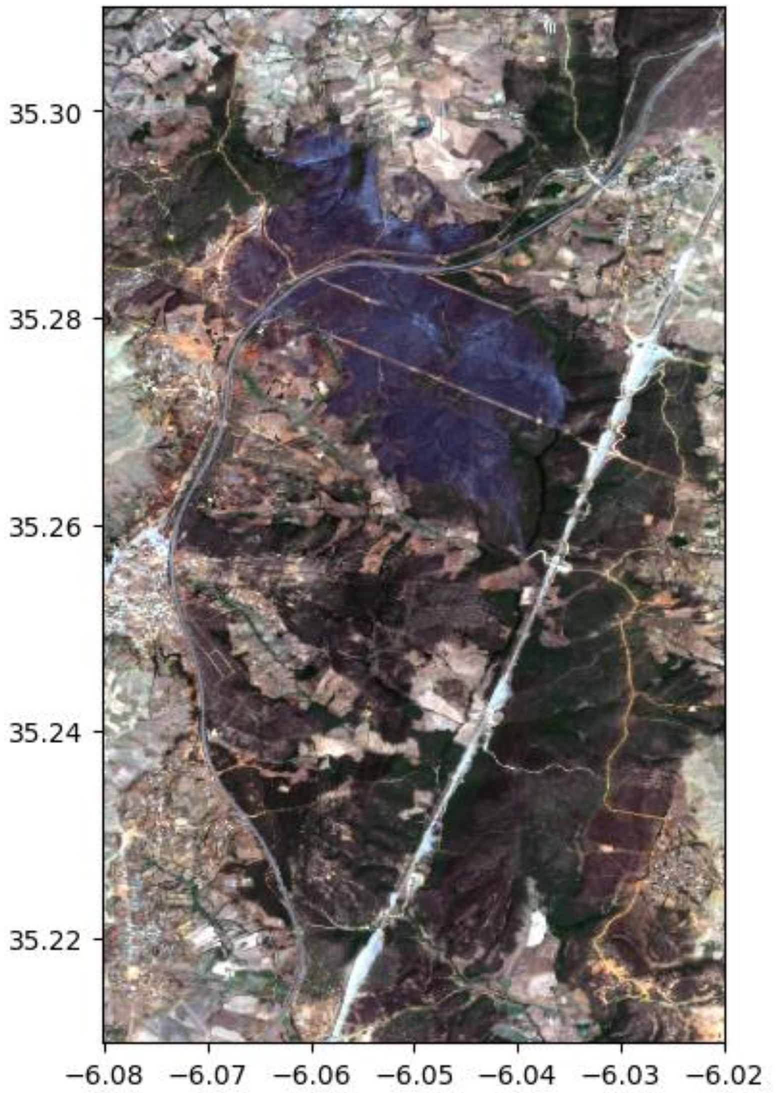
Showing the after fire image of Larache as a colour composite

```
'''
----------------------------------------------------------------
The following block of code originates and is modified from:

Balzter, H. (2023)
Materials for GY7709 masters computer classes.
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-
top%2FGY7709%5FSatellite%5FData%5FAnalysis%
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper-
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1
Downloaded 1 February 2023
----------------------------------------------------------------
'''
# getting list of all tiff files in the directory
allfiles = [f for f in listdir(downloaddir) if isfile(join(downloaddir, f))]
print(allfiles)
```

```
# selecting the file for visualisation
thisfile = allfiles[0]
print(thisfile)
['s2.tif']
s2.tif
```

True Colour composite of after fire image of Larache

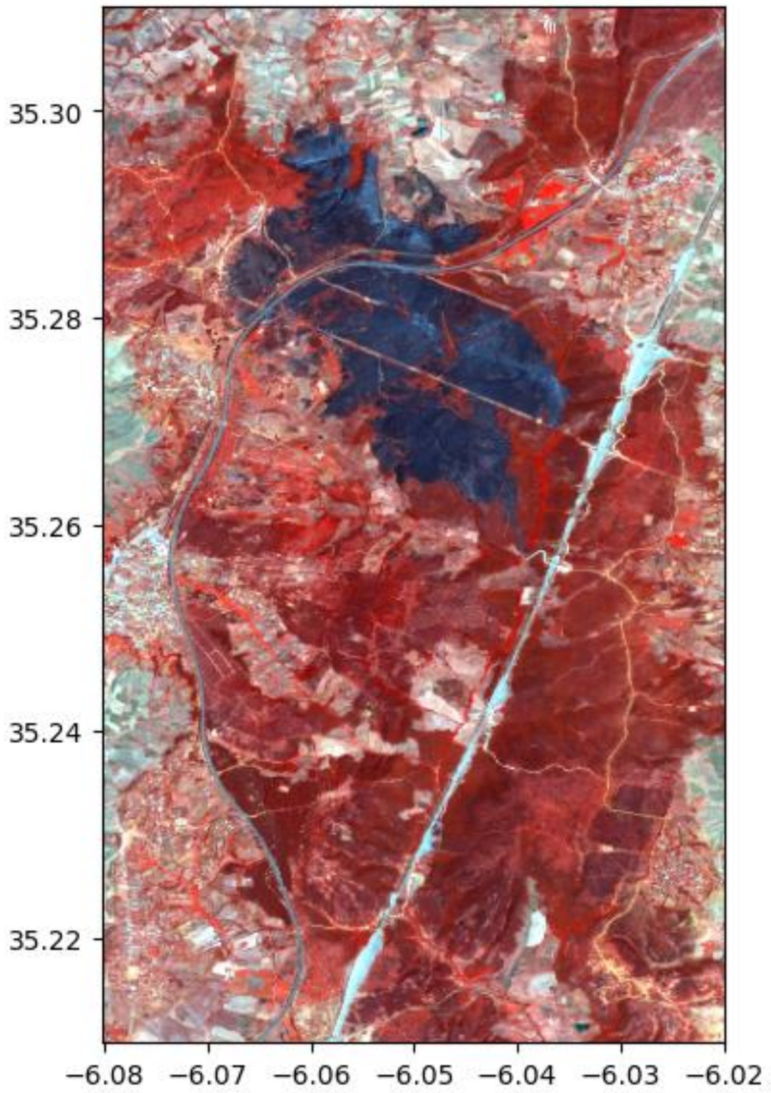
```
'''
--------------------------------------------------------------
The following block of code originates and is modified from:

Balzter, H. (2023)
Materials for GY7709 masters computer classes.
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-
top%2FGY7709%5FSatellite%5FData%5FAnalysis%
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper-
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1
Downloaded 1 February 2023
--------------------------------------------------------------
'''
# creating a figure with 2x3 subplots
fig, (ax1) = plt.subplots(1,1, figsize=(7,7))
fig.patch.set_facecolor('white')

# the downloaded file is float32 data format
# for plotting,  uint8 data format is needed

# plotting the image with full extent
pygge.easy_plot(thisfile, ax=ax1, bands=[3,2,1], percentiles=[0,99])
WARNING:rasterio._env:CPLE_AppDefined in s2.tif: TIFFReadDirectory:Sum of Photometric t
ype-related color channels and ExtraSamples doesn't match SamplesPerPixel. Defining non
-color channels as ExtraSamples.
WARNING:rasterio._env:CPLE_AppDefined in TIFFReadDirectory:Sum of Photometric type-rela
ted color channels and ExtraSamples doesn't match SamplesPerPixel. Defining non-color c
hannels as ExtraSamples.
```



False Colour Composite of after fire image of Larache

```
'''
--------------------------------------------------------------
The following block of code originates and is modified from:

Balzter, H. (2023)
Materials for GY7709 masters computer classes.
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-
top%2FGY7709%5FSatellite%5FData%5FAnalysis%
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper-
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1
```

```
Downloaded 1 February 2023
--------------------------------------------------------------
'''
# creating a figure with 2x3 subplots
fig, (ax1) = plt.subplots(1,1, figsize=(7,7))
fig.patch.set_facecolor('white')

# the downloaded file is float32 data format
# for plotting, we need uint8 data format

# plotting the image with full extent
pygge.easy_plot(thisfile, ax=ax1, bands=[4,2,1], percentiles=[0,99])
WARNING:rasterio._env:CPLE_AppDefined in s2.tif: TIFFReadDirectory:Sum of Photometric t
ype-related color channels and ExtraSamples doesn't match SamplesPerPixel. Defining non
-color channels as ExtraSamples.
WARNING:rasterio._env:CPLE_AppDefined in TIFFReadDirectory:Sum of Photometric type-rela
ted color channels and ExtraSamples doesn't match SamplesPerPixel. Defining non-color c
hannels as ExtraSamples.
```



Warping the downloaded after fire image composite of larache into another map projection

23. The coordinate reference system (CRS) of the downloaded image composite is not in the UK national map projection. It is reprojected.

```

'''
--------------------------------------------------------------
The following block of code originates and is modified from:

Balzter, H. (2023)
Materials for GY7709 masters computer classes.
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15
```

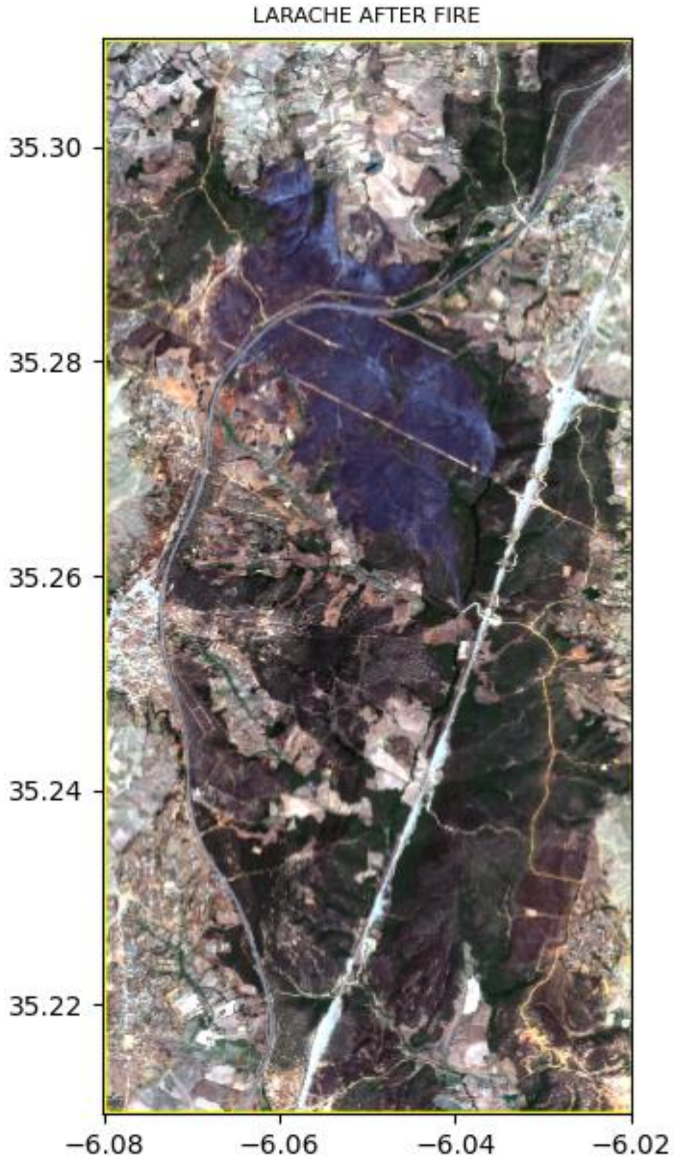
```
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-
top%2FGY7709%5FSatellite%5FData%5FAnalysis%
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper-
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1
Downloaded 1 February 2023
--------------------------------------------------------------
'''
# printing the EPSG code of our shapefile into which we want to reproject the TCI im-
ages
print("Reprojecting image to EPSG projection ", epsg)

# making a file name for our new file
warpfile = thisfile.split(sep='.')[0] + '_warped.tif'
print("We are in this directory: ")
!pwd
print("Input file: ", thisfile)
print("Output file: ", warpfile)

# calling the easy_warp function
tmp = pygge.easy_warp(thisfile, warpfile, epsg)
Reprojecting image to EPSG projection  4326
We are in this directory:
/content/work/download
WARNING:rasterio._env:CPLE_AppDefined in s2.tif: TIFFReadDirectory:Sum of Photometric t
ype-related color channels and ExtraSamples doesn't match SamplesPerPixel. Defining non
-color channels as ExtraSamples.
WARNING:rasterio._env:CPLE_AppDefined in TIFFReadDirectory:Sum of Photometric type-rela
ted color channels and ExtraSamples doesn't match SamplesPerPixel. Defining non-color c
hannels as ExtraSamples.
Input file:  s2.tif
Output file:  s2_warped.tif
Creating warped file:s2_warped.tif
```

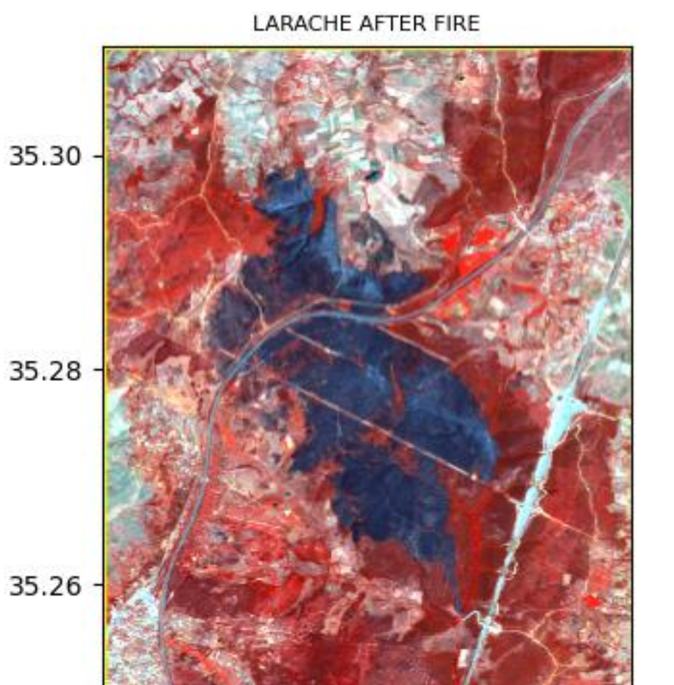
Plotting the shapefile on top of the raster

24. to see the locations of our polygons on top of our image composite,We do that with the Geopandas library.

```
'''
--------------------------------------------------------------
The following block of code originates and is modified from:
```

```
Balzter, H. (2023)
Materials for GY7709 masters computer classes.
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-
top%2FGY7709%5FSatellite%5FData%5FAnalysis%
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper-
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1
Downloaded 1 February 2023
--------------------------------------------------------------
'''
# creating a figure with subplots
fig, ax = plt.subplots(2,1, figsize=(10,16))
fig.patch.set_facecolor('white')

# plotting the image with full extent
pygge.easy_plot(warpfile, ax=ax[0], percentiles=[0,98], bands=[3,2,1],
                shapefile=shapefile, fillcolor="none", linecolor="yellow",
                title="LARACHE AFTER FIRE")

# plotting the image with full extent
pygge.easy_plot(warpfile, ax=ax[1], percentiles=[0,98], bands=[4,2,1],
                shapefile=shapefile, fillcolor="none", linecolor="yellow",
                title="LARACHE AFTER FIRE")
```


LARACHE AFTER FIRE
35.30
35.28
35.26
35.24
35.22
−6.08
−6.06
−6.04
−6.02


LARACHE AFTER FIRE
35.30
35.28
35.26

READING IN DOWNLOADED TIF FILE OF LARACHE BEFORE FIRE AND GETTING ITS METADATA

```
'''
---------------------------------------------------------------
The following block of code originates and is modified from:

Balzter, H. (2023)
Materials for GY7709 masters computer classes.
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-
top%2FGY7709%5FSatellite%5FData%5FAnalysis%
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper-
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1
Downloaded 1 February 2023
---------------------------------------------------------------
'''
# reading in the warped before fire downloaded image of larache and checking its cordi-
nate reference system
larache_before = rasterio.open(join(rootdir +  "/larache_before/be-
fore_fire_warped.tif"))
src1=larache_before
d1=src1.crs.to_wkt()
print(type(d1))
pprint(d1)
<class 'str'>
('GEOGCS["WGS 84",DATUM["WGS_1984",SPHEROID["WGS '
 '84",6378137,298.257223563,AUTHORITY["EPSG","7030"]],AUTHORITY["EPSG","6326"]],PRIMEM[
"Greenwich",0,AUTHORITY["EPSG","8901"]],UNIT["degree",0.0174532925199433,AUTHORITY["EPS
G","9122"]],AXIS["Latitude",NORTH],AXIS["Longitude",EAST],AUTHORITY["EPSG","4326"]]')
```

READING IN DOWNLOADED TIF FILE OF LARACHE AFTER FIRE AND GETTING ITS METADATA

```
'''
---------------------------------------------------------------
```

```
The following block of code originates and is modified from:

Balzter, H. (2023)
Materials for GY7709 masters computer classes.
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-
top%2FGY7709%5FSatellite%5FData%5FAnalysis%
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper-
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1
Downloaded 1 February 2023
--------------------------------------------------------------
'''
# reading in the warped after fire downloaded image of larache and checking its cordi-
nate system
larache_after = rasterio.open(join(rootdir +  "/larache_after/after_fire_warped.tif"))
src2=larache_after
d2=src2.crs.to_wkt()
print(type(d2))
pprint(d2)
<class 'str'>
('GEOGCS["WGS 84",DATUM["WGS_1984",SPHEROID["WGS '
 '84",6378137,298.257223563,AUTHORITY["EPSG","7030"]],AUTHORITY["EPSG","6326"]],PRIMEM[
"Greenwich",0,AUTHORITY["EPSG","8901"]],UNIT["degree",0.0174532925199433,AUTHORITY["EPS
G","9122"]],AXIS["Latitude",NORTH],AXIS["Longitude",EAST],AUTHORITY["EPSG","4326"]]')
```

CHECKING IF THE BEFORE AND AFTER DATASETS ARE OF SAME SHAPE

```
# checking if the before and after image data are the same shape?
larache_before.shape == larache_after.shape

True
```

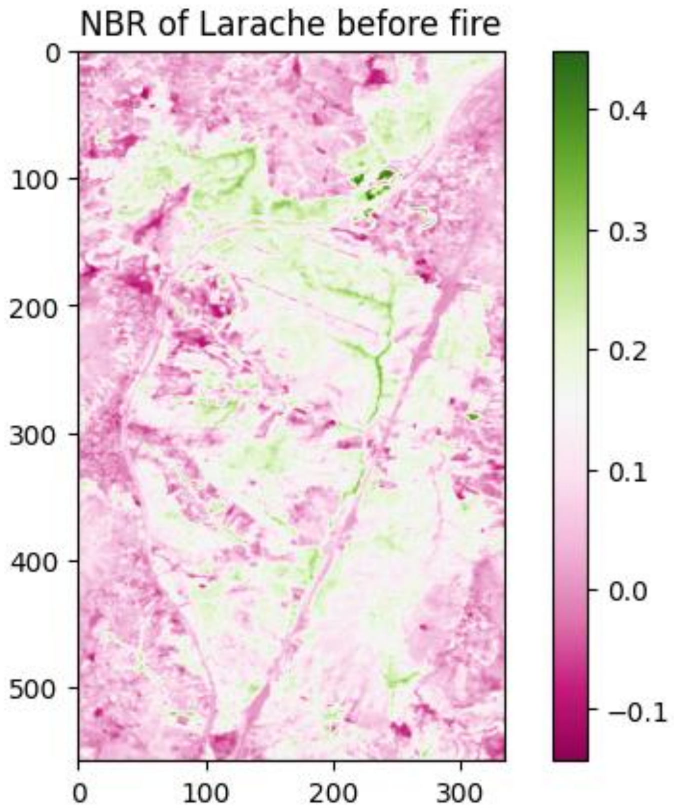
CALCULATION OF NBR FOR THE BEFORE FIRE IMAGE OF LARACHE

```
'''
--------------------------------------------------------------
The following block of code originates and is modified from:
```

```
Balzter, H. (2023)
Materials for GY7709 masters computer classes.
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-
top%2FGY7709%5FSatellite%5FData%5FAnalysis%
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper-
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1
Downloaded 1 February 2023
--------------------------------------------------------------
'''
swir_before = larache_before.read(5) # band in position 5 in our stacked image i.e SWIR
nir_before = larache_before.read(4) # band in position 4 in our stacked image i.e NIR

# calculation of NBR before the fire as flaoting point array
nbr_before = (nir_before.astype(float) - swir_before.astype(float)) / (nir_be-
fore.astype(float) + swir_before.astype(float))


# Ignoring division by zero
np.seterr(divide='ignore', invalid='ignore')

# printing some image statistics, ignoring missing values (nan)
print("minimum NBR_before = ", np.nanmin(nbr_before))
print("mean NBR_before = ", np.nanmean(nbr_before))
print("maximum NBR_before = ", np.nanmax(nbr_before))
print("standard deviation = ", np.nanstd(nbr_before))
minimum NBR_before =  -0.14298390392538413
mean NBR_before =  0.10239791290152904
maximum NBR_before =  0.44895833342572644
standard deviation =  0.0735262823032699
```

PLOT OF NBR BEFORE THE FIRE

```
plt.imshow(nbr_before, cmap='PiYG')     # colours to use  include Purple and Green
plt.colorbar()
plt.title('NBR of Larache before fire')  # title of plot
plt.show()                               # plot
```



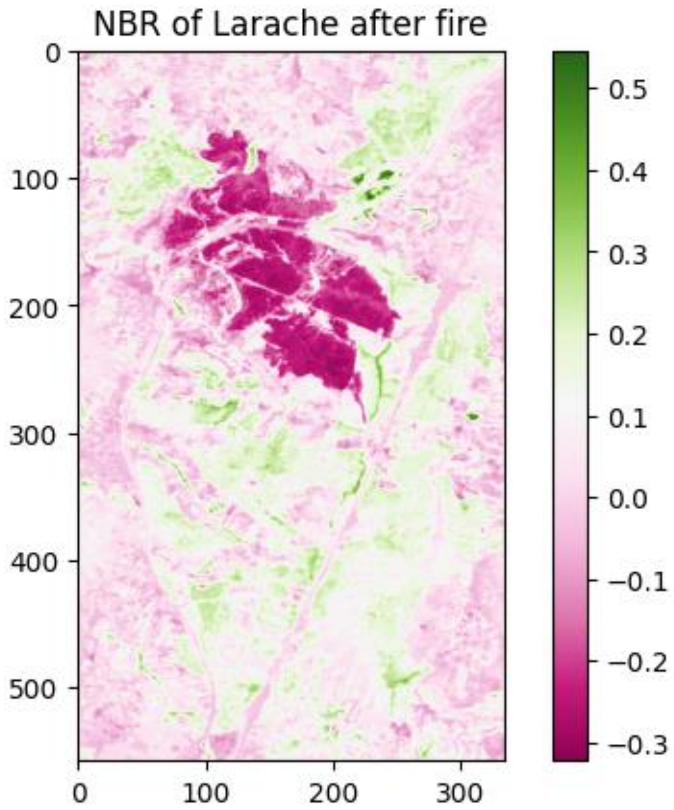
CALCULATION OF NBR FOR THE AFTER FIRE IMAGE OF LARACHE

```
'''
--------------------------------------------------------------
The following block of code originates and is modified from:

Balzter, H. (2023)
Materials for GY7709 masters computer classes.
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-
top%2FGY7709%5FSatellite%5FData%5FAnalysis%
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper-
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1
Downloaded 1 February 2023
--------------------------------------------------------------
'''
swir_after = larache_after.read(5) # band in position 5 in our stacked image i.e SWIR
nir_after = larache_after.read(4) # band in position 4 in our stacked image i.e NIR
```

```
# calculation of NBR after the fire as flaoting point array
nbr_after = (nir_after.astype(float) - swir_after.astype(float)) / (nir_after.astype(float) + swir_after.astype(float))

# Ignoring division by zero
np.seterr(divide='ignore', invalid='ignore')

# printing some image statistics, ignoring missing values (nan)
print("minimum NBR_after = ", np.nanmin(nbr_after))
print("mean NBR_after = ", np.nanmean(nbr_after))
print("maximum NBR_after = ", np.nanmax(nbr_after))
print("standard deviation = ", np.nanstd(nbr_after))
minimum NBR_after =  -0.3224225354386313
mean NBR_after =  0.049028595859026615
maximum NBR_after =  0.5460599474035327
standard deviation =  0.11624440093712922
```

PLOT OF NBR AFTER THE FIRE

```
plt.imshow(nbr_after, cmap='PiYG') # Purple and Green selected as colours in plot
plt.colorbar()
plt.title('NBR of Larache after fire')  # title of plot
plt.show()                              # display plot or map
```



CALCULATION OF dNBR AS DIFFERENCE OF BEFORE AND AFTER NBR RASTERS

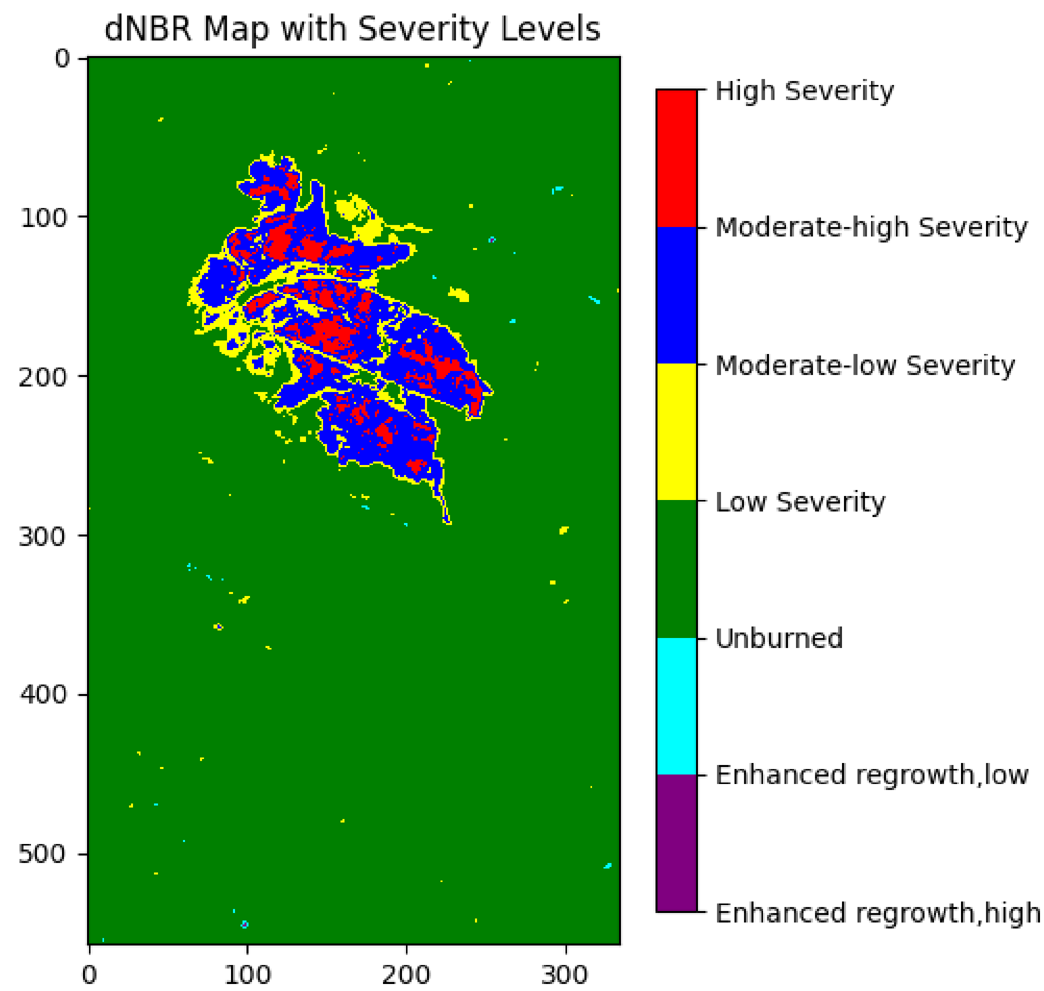
```
# finding the difference between the NBR before and after the fire
dNBR = nbr_before - nbr_after

'''
--------------------------------------------------------------
The following block of code originates and is modified from:

Balzter, H. (2023)
Materials for GY7709 masters computer classes.
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-
top%2FGY7709%5FSatellite%5FData%5FAnalysis%
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper-
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1
Downloaded 1 February 2023
--------------------------------------------------------------
'''
```

```
# printing some image statistics, ignoring missing values (nan)
print("minimum dNBR = ", np.nanmin(dNBR))
print("mean dNBR = ", np.nanmean(dNBR))
print("maximum dNBR = ", np.nanmax(dNBR))
print("standard deviation = ", np.nanstd(dNBR))
minimum dNBR =  -0.363571844669598
mean dNBR =  0.05336931704250243
maximum dNBR =  0.5638484811508442
standard deviation =  0.10752007563079775
```

MAP SHOWING AREAS WHERE THE FIRE DAMAGE HAS BEEN GREATEST

```
# Defining the dNBR values and the corresponding severity levels
dNBR

levels = [-0.5, -0.25, -0.1, 0.1, 0.27,0.44,0.66]
labels = ['Enhanced regrowth,high','Enhanced regrowth,low','Unburned', 'Low Severity',
'Moderate-low Severity', 'Moderate-high Severity', 'High Severity']

# Creating a custom color map with 5 colors
cmap = colors.ListedColormap(['purple','cyan','green', 'yellow', 'blue', 'darkred',
'red'])

# Setting the boundaries and norm of the color map
bounds = [-0.5, -0.25, -0.1, 0.1, 0.27,0.44,0.66]
norm = colors.BoundaryNorm(bounds, cmap.N)

# Creating a figure and axis object
fig, ax = plt.subplots(figsize=(6, 6))

# Plotting the dNBR using the custom color map and norm
im = ax.imshow(dNBR, cmap=cmap, norm=norm)

# Adding a color bar with custom tick labels
cbar = fig.colorbar(im, ax=ax, fraction=0.046, pad=0.04, boundaries=bounds, ticks=lev-
els)
cbar.ax.set_yticklabels(labels)

# Setting the title and axis labels
ax.set_title('dNBR Map with Severity Levels')
```

```
plt.show()
```



CALCULATION OF AREAS OF BURNED CLASSES USING dNBR

```
# Define the area of each pixel in hectares assuming a spatial resolution of 20 meters
pixel_area = 0.0004

# Calculate the number of pixels in each class or label
regrowth_high_pixels = np.sum((dNBR >= -0.5) & (dNBR < -0.251))
regrowth_low_pixels = np.sum((dNBR >= -0.25) & (dNBR < -0.101))
unburned_pixels = np.sum((dNBR >= -0.100) & (dNBR < 0.99))
low_severity_pixels = np.sum((dNBR >= 0.100) & (dNBR < 0.269))
mod_low_severity_pixels = np.sum((dNBR >= 0.270) & (dNBR < 0.439))
mod_high_severity_pixels = np.sum((dNBR >= 0.44) & (dNBR < 0.659))
high_severity_pixels = np.sum((dNBR >= 0.660) & (dNBR < 1.300))
```

```
# Calculate the area of each class or label in hectares
regrowth_high_area = regrowth_high_pixels * pixel_area
regrowth_low_area = regrowth_low_pixels * pixel_area
unburned_area = unburned_pixels * pixel_area
low_severity_area = low_severity_pixels * pixel_area
mod_low_severity_area = mod_low_severity_pixels * pixel_area
mod_high_severity_area = mod_high_severity_pixels * pixel_area
high_severity_area = high_severity_pixels * pixel_area

# Print the area of each class or label
print('Enhanced regrowth,high area:', regrowth_high_area, 'hectares')
print('Enhanced regrowth, low area:', regrowth_low_area, 'hectares')
print('Unburned area:', unburned_area, 'hectares')
print('Low severity area:', low_severity_area, 'hectares')
print('Moderate-low severity area:', mod_low_severity_area, 'hectares')
print('Moderate-high severity area:', mod_high_severity_area, 'hectares')
print('High severity area:', high_severity_area, 'hectares')
Enhanced regrowth,high area: 0.0036000000000000003 hectares
Enhanced regrowth, low area: 0.0664 hectares
Unburned area: 74.6992 hectares
Low severity area: 2.0148 hectares
Moderate-low severity area: 4.2816 hectares
Moderate-high severity area: 1.4420000000000002 hectares
High severity area: 0.0 hectares
```

CALCULATION OF AVERAGE NBR2 FOR BURNED POLYGON

```
# Opening the TIF file and the shapefile of burned polygon
with rasterio.open('/content/drive/MyDrive/satelliteCW1/larache_after/after_fire_warped.tif') as src:
    shapes = gpd.read_file('/content/drive/MyDrive/satelliteCW1/burned_layers/POLYGON.shp')

    # Masking the TIF file using the shapefile
    masked_data, _ = rasterio.mask.mask(src, shapes.geometry, crop=True)

    # Calculating the NBR2_burned
    # the NIR band has index 3 and SWIR band has index 4
    NBR2_burned = (masked_data[3] - masked_data[4]) / (masked_data[3] + masked_data[4])

    # Ignoring division by zero
```

```
np.seterr(divide='ignore', invalid='ignore')

# print average NBR2 for burned poloygon, ignoring missing values (nan)

print("average NBR2_burned = ", np.nanmean(NBR2_burned))
average NBR2_burned =  -0.13870606
```

CALCULATION OF AVERAGE NBR1 FOR BURNED POLYGON

```
# Opening the TIF file and the shapefile of burned polygon
with rasterio.open('/content/drive/MyDrive/satelliteCW1/larache_before/be-
fore_fire_warped.tif') as src:
    shapes = gpd.read_file('/content/drive/MyDrive/satelliteCW1/burned_layers/POLY-
GON.shp')

    # Masking the TIF file using the shapefile
    masked_data, _ = rasterio.mask.mask(src, shapes.geometry, crop=True)

    # Calculating the NBR1_burned
    # the NIR  band has index 3 and SWIR band has index 4
    NBR1_burned = (masked_data[3] - masked_data[4]) / (masked_data[3] + masked_data[4])

    # Ignoring division by zero
np.seterr(divide='ignore', invalid='ignore')

# printing average NBR1 for burned polygon, ignoring missing values (nan)

print("average NBR1_burned = ", np.nanmean(NBR1_burned))
average NBR1_burned =  0.1715795
```

25. CALCULATION OF dNBR FOR BURNED POLYGON

```
#  finding the difference between the NBR before and after the fire
dNBR_burned = NBR1_burned - NBR2_burned

# printing average dNBR for burned polygon, ignoring missing values (nan)

print("average dNBR_burned = ", np.nanmean(dNBR_burned))
average dNBR_burned =  0.3102856
```

CALCULATION OF AVERAGE NBR1 FOR UNBURNED POLYGON

```
# Opening the TIF file and the shapefile of unburned polygon
with rasterio.open('/content/drive/MyDrive/satelliteCW1/larache_before/be-
fore_fire_warped.tif') as src:
    shapes = gpd.read_file('/content/drive/MyDrive/satelliteCW1/unburned_layers/POLY-
GON.shp')

    # Masking the TIF file using the shapefile
    masked_data, _ = rasterio.mask.mask(src, shapes.geometry, crop=True)

    # Calculating the NBR1_unburned
    # the NIR band has index 3 and SWIR band has index 4
    NBR1_unburned = (masked_data[3] - masked_data[4]) / (masked_data[3] +
masked_data[4])

    # Ignoring division by zero
np.seterr(divide='ignore', invalid='ignore')

# printing average NBR1 for unburned poloygon, ignoring missing values (nan)

print("average NBR1_unburned = ", np.nanmean(NBR1_unburned))
average NBR1_unburned =  0.17309634
```

CALCULATION OF AVERAGE NBR2 FOR UNBURNED POLYGON

```
# Opening the TIF file and the shapefile of unburned polygon
with rasterio.open('/content/drive/MyDrive/satelliteCW1/larache_after/af-
ter_fire_warped.tif') as src:
    shapes = gpd.read_file('/content/drive/MyDrive/satelliteCW1/unburned_layers/POLY-
GON.shp')

    # Masking the TIF file using the shapefile
    masked_data, _ = rasterio.mask.mask(src, shapes.geometry, crop=True)

    # Calculating the NBR2_unburned
    # the NIR band has index 3 and SWIR band has index 4
    NBR2_unburned = (masked_data[3] - masked_data[4]) / (masked_data[3] +
masked_data[4])

    # Ignoring division by zero
np.seterr(divide='ignore', invalid='ignore')
```

```
# printing average NBR2 for unburned poloygon, ignoring missing values (nan)

print("average NBR2_unburned = ", np.nanmean(NBR2_unburned))
average NBR2_unburned =  0.17163754
```

CALCULATION OF dNBR FOR UNBURNED POLYGON

```
# finding the difference between the NBR before and after the fire
dNBR_unburned = NBR1_unburned - NBR2_unburned

# printing average dNBR for burned polygon, ignoring missing values (nan)

print("average dNBR_unburned = ", np.nanmean(dNBR_unburned))
average dNBR_unburned =  0.0014588113
```

CALCULATION OF NDVI FOR THE BEFORE FIRE IMAGE OF LARACHE

```
'''
--------------------------------------------------------------
The following block of code originates and is modified from:

Balzter, H. (2023)
Materials for GY7709 masters computer classes.
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-
top%2FGY7709%5FSatellite%5FData%5FAnalysis%
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper-
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1
Downloaded 1 February 2023
--------------------------------------------------------------
'''
r_before = larache_before.read(3) # band 3 in our stacked image is Red
nir_before = larache_before.read(4) # band 4 in our stacked image is NIR

# calculation of NDVI before the fire as flaoting point array
NDVI_before = (nir_before.astype(float) - r_before.astype(float)) / (nir_be-
fore.astype(float) + r_before.astype(float))
```

```
# Ignore division by zero
np.seterr(divide='ignore', invalid='ignore')

# print some image statistics, ignoring missing values (nan)
print("minimum NDVI_before = ", np.nanmin(NDVI_before))
print("mean NDVI_before = ", np.nanmean(NDVI_before))
print("maximum NDVI_before = ", np.nanmax(NDVI_before))
print("standard deviation = ", np.nanstd(NDVI_before))
minimum NDVI_before =  -0.0031236251459653844
mean NDVI_before =  0.16960801456931313
maximum NDVI_before =  0.44798869776436046
standard deviation =  0.053921411038247144
```

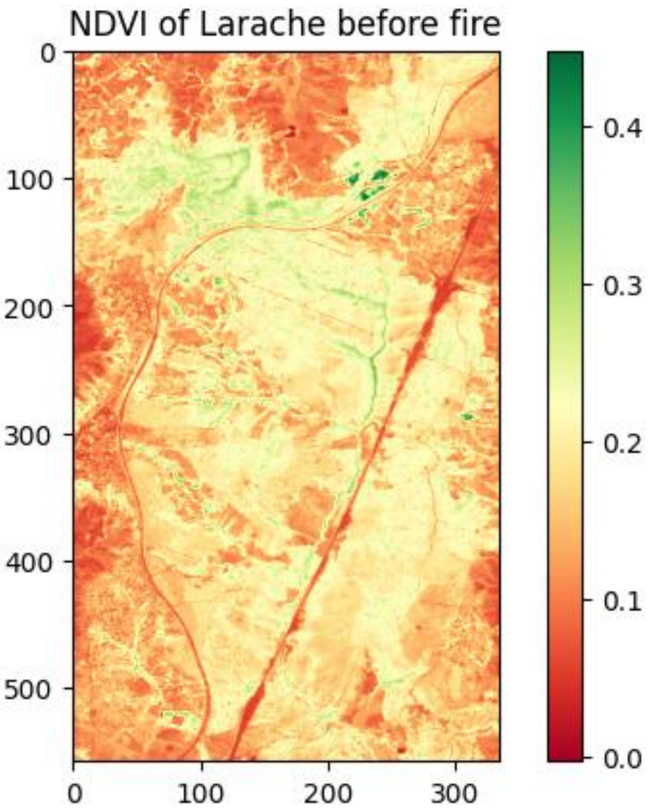
CALCULATION OF NDVI FOR THE AFTER FIRE IMAGE OF LARACHE

```
'''
--------------------------------------------------------------
The following block of code originates and is modified from:

Balzter, H. (2023)
Materials for GY7709 masters computer classes.
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-
top%2FGY7709%5FSatellite%5FData%5FAnalysis%
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper-
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1
Downloaded 1 February 2023
--------------------------------------------------------------
'''
r_after = larache_after.read(3) # band 3 in our stacked image is Red
nir_after = larache_after.read(4) # band 4 in our stacked image is NIR

# calculation of NDVI after the fire as flaoting point array
NDVI_after = (nir_after.astype(float) - r_after.astype(float)) / (nir_af-
ter.astype(float) + r_after.astype(float))

# Ignoring division by zero
np.seterr(divide='ignore', invalid='ignore')
```

```
# print some image statistics, ignoring missing values (nan)
print("minimum NDVI_after = ", np.nanmin(NDVI_after))
print("mean NDVI_after = ", np.nanmean(NDVI_after))
print("maximum NDVI_after = ", np.nanmax(NDVI_after))
print("standard deviation = ", np.nanstd(NDVI_after))
minimum NDVI_after =  -0.06513785137467486
mean NDVI_after =  0.17277971166410844
maximum NDVI_after =  0.5750123655099424
standard deviation =  0.07998356629775627
```

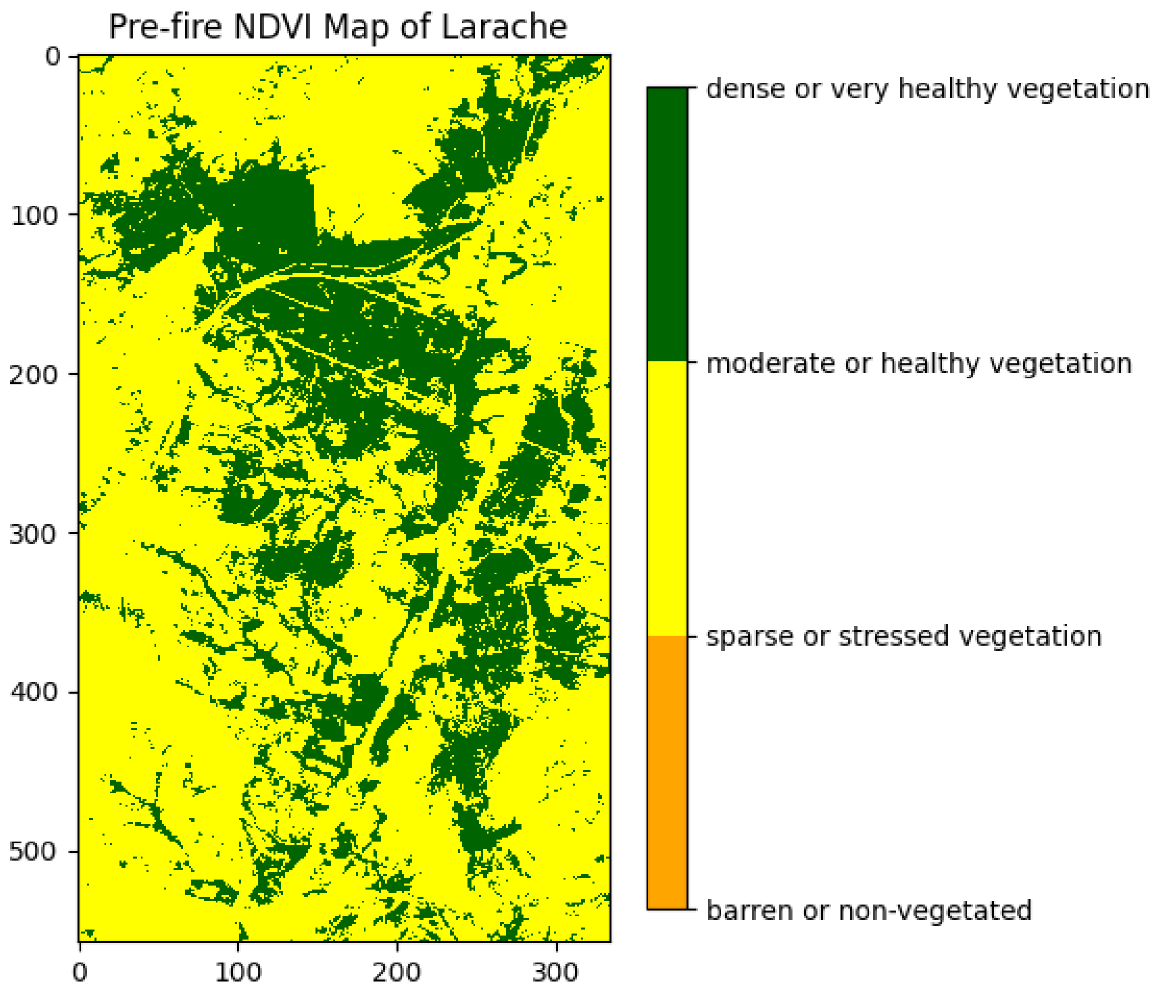
MAPS OF NDVI BEFORE THE FIRE

```
plt.imshow(NDVI_before, cmap='RdYlGn') # selection of Red to Green colours
plt.colorbar()                          # colour bar
plt.title('NDVI of Larache before fire')  # map title
plt.show()                              # showing plot
```



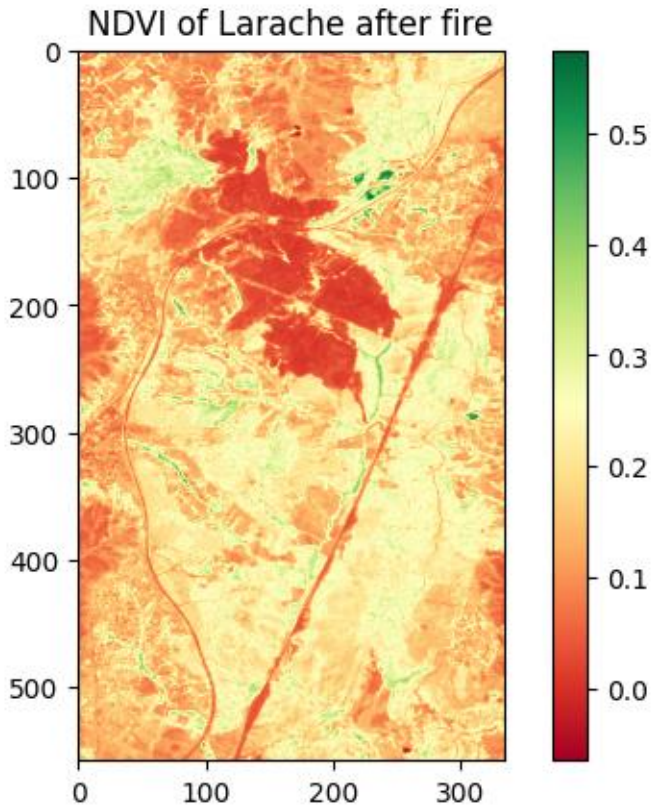

```
# Defining the pre-fire NDVI  levels
NDVI_before

levels = [-1, 0, 0.2, 0.5]
```

```
labels = ['barren or non-vegetated','sparse or stressed vegetation','moderate or
healthy vegetation', 'dense or very healthy vegetation']

# Creating a custom color map with 5 colors
cmap = colors.ListedColormap(['orange', 'yellow', 'lightgreen', 'darkgreen'])

# Setting the boundaries and norm of the color map
bounds = [-1, 0, 0.2, 0.5]
norm = colors.BoundaryNorm(bounds, cmap.N)

# Creating a figure and axis object
fig, ax = plt.subplots(figsize=(6, 6))

# Plotting the pre-fire NDVI using the custom color map and norm
im = ax.imshow(NDVI_before, cmap=cmap, norm=norm)

# Adding a color bar with custom tick labels
cbar = fig.colorbar(im, ax=ax, fraction=0.046, pad=0.04, boundaries=bounds, ticks=lev-
els)
cbar.ax.set_yticklabels(labels)

# Setting the title and axis labels
ax.set_title('Pre-fire NDVI Map of Larache ')

plt.show()
```



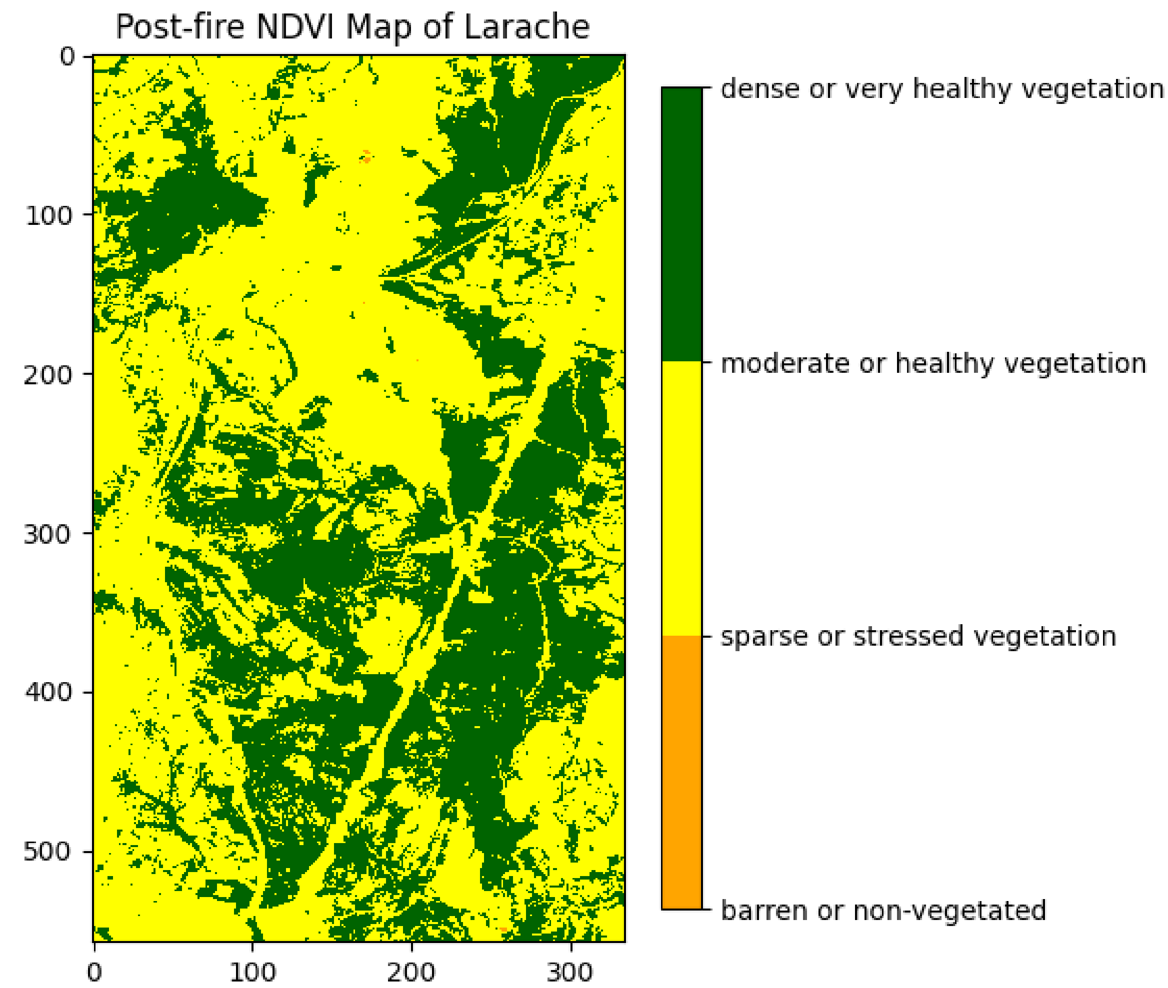
26. MAPS OF NDVI AFTER THE FIRE

```
plt.imshow(NDVI_after, cmap='RdYlGn')       # Red to Green colour selected
plt.colorbar()
plt.title('NDVI of Larache after fire')     # title
plt.show()
```




```
# Defining the post-fire NDVI  levels
NDVI_after

levels = [-1, 0, 0.2, 0.5]
labels = ['barren or non-vegetated','sparse or stressed vegetation','moderate or
healthy vegetation', 'dense or very healthy vegetation']

# Creating a custom color map with 5 colors
cmap = colors.ListedColormap(['orange', 'yellow', 'lightgreen', 'darkgreen'])

# Setting the boundaries and norm of the color map
bounds = [-1, 0, 0.2, 0.5]
norm = colors.BoundaryNorm(bounds, cmap.N)

# Creating a figure and axis object
fig, ax = plt.subplots(figsize=(6, 6))

# Plotting the post-fire NDVI using the custom color map and norm
im = ax.imshow(NDVI_after, cmap=cmap, norm=norm)

# Adding a color bar with custom tick labels
```

```
cbar = fig.colorbar(im, ax=ax, fraction=0.046, pad=0.04, boundaries=bounds, ticks=levels)
cbar.ax.set_yticklabels(labels)

# Setting the title and axis labels
ax.set_title('Post-fire NDVI Map of Larache ')

plt.show()
```



CALCULATING THE NDVI DIFFERENCE (dNDVI)

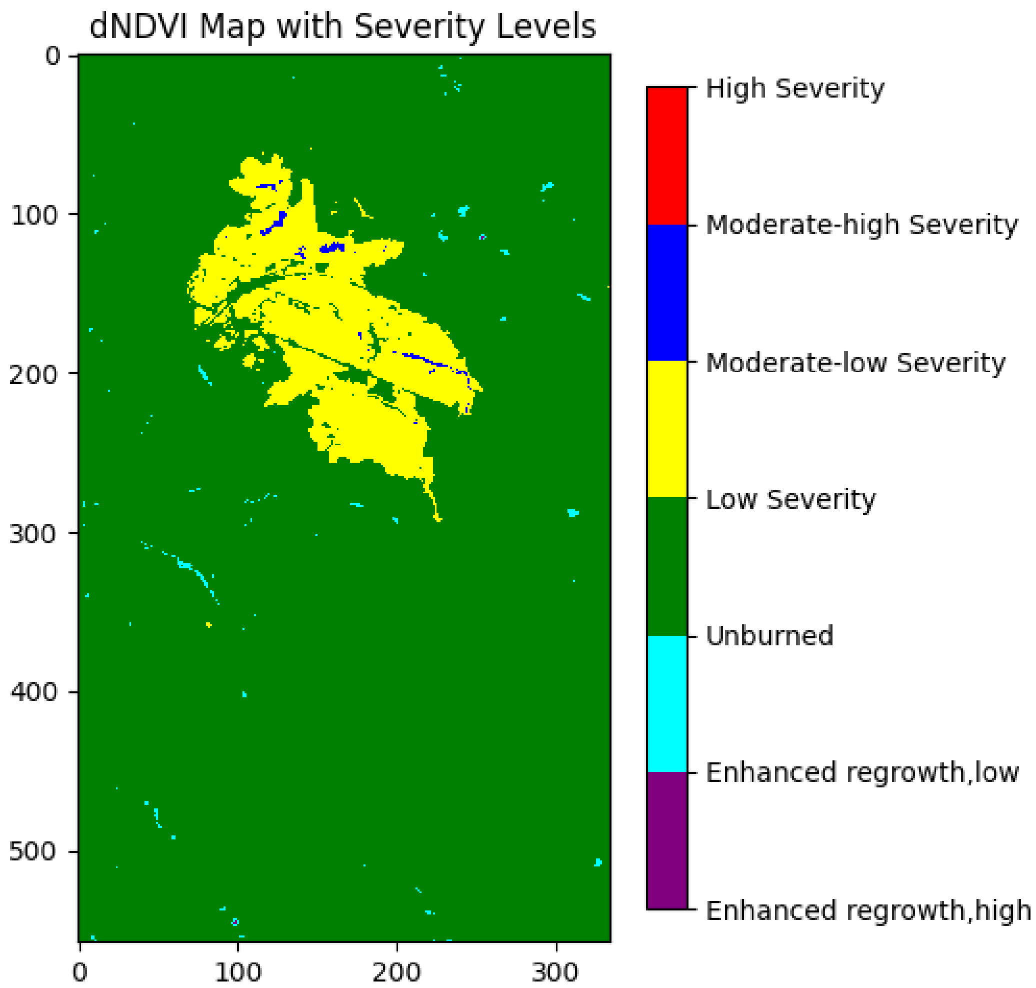
```
# finding the difference between the NDVI before and after the fire
dNDVI = NDVI_before - NDVI_after

'''
--------------------------------------------------------------
The following block of code originates and is modified from:
```

```
Balzter, H. (2023)
Materials for GY7709 masters computer classes.
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-
top%2FGY7709%5FSatellite%5FData%5FAnalysis%
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper-
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1
Downloaded 1 February 2023
--------------------------------------------------------------
'''

# printing some image statistics, ignoring missing values (nan)
print("minimum dNDVI = ", np.nanmin(dNDVI))
print("mean dNDVI = ", np.nanmean(dNDVI))
print("maximum dNDVI = ", np.nanmax(dNDVI))
print("standard deviation = ", np.nanstd(dNDVI))
minimum dNDVI =  -0.3098725340997325
mean dNDVI =  -0.0031716970947953077
maximum dNDVI =  0.33930637628747506
standard deviation =  0.06478898685069448
```

dNDVI based BURN SEVERITY MAP

```
# Defining the dNDVI values and the corresponding severity levels
dNDVI

levels = [-0.5, -0.25, -0.1, 0.1, 0.27,0.44,0.66]
labels = ['Enhanced regrowth,high','Enhanced regrowth,low','Unburned', 'Low Severity',
'Moderate-low Severity', 'Moderate-high Severity', 'High Severity']

# Creating a custom color map with 5 colors
cmap = colors.ListedColormap(['purple','cyan','green', 'yellow', 'blue', 'darkred',
'red'])

# Setting the boundaries and norm of the color map
bounds = [-0.5, -0.25, -0.1, 0.1, 0.27,0.44,0.66]
norm = colors.BoundaryNorm(bounds, cmap.N)

# Creating a figure and axis object
```

```
fig, ax = plt.subplots(figsize=(6, 6))

# Plotting the dNDVI using the custom color map and norm
im = ax.imshow(dNDVI, cmap=cmap, norm=norm)

# Adding a color bar with custom tick labels
cbar = fig.colorbar(im, ax=ax, fraction=0.046, pad=0.04, boundaries=bounds, ticks=lev-
els)
cbar.ax.set_yticklabels(labels)

# Setting the title and axis labels
ax.set_title('dNDVI Map with Severity Levels')

plt.show()
```

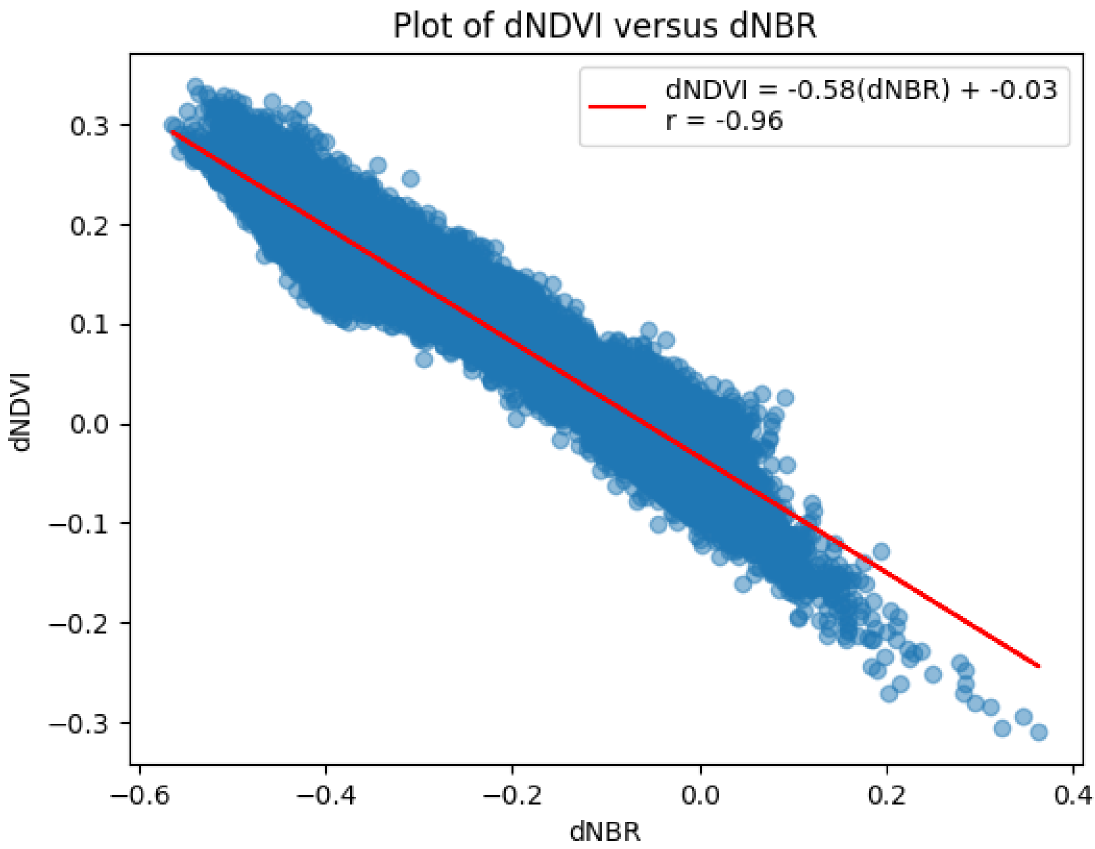


CALCULATION OF BURNED AREAS BASED ON dNDVI

```
# Define the area of each pixel in hectares assuming a spatial resolution of 20 meters
pixel_area = 0.0004

# Calculate the number of pixels in each class or label
regrowth_high_pixels = np.sum((dNDVI >= -0.5) & (dNDVI < -0.251))
regrowth_low_pixels = np.sum((dNDVI >= -0.25) & (dNDVI < -0.101))
unburned_pixels = np.sum((dNDVI >= -0.100) & (dNDVI < 0.99))
low_severity_pixels = np.sum((dNDVI >= 0.100) & (dNDVI < 0.269))
mod_low_severity_pixels = np.sum((dNDVI >= 0.270) & (dNDVI < 0.439))
mod_high_severity_pixels = np.sum((dNDVI >= 0.44) & (dNDVI < 0.659))
high_severity_pixels = np.sum((dNDVI >= 0.660) & (dNDVI < 1.300))

# Calculate the area of each class or label in hectares
regrowth_high_area = regrowth_high_pixels * pixel_area
regrowth_low_area = regrowth_low_pixels * pixel_area
unburned_area = unburned_pixels * pixel_area
low_severity_area = low_severity_pixels * pixel_area
mod_low_severity_area = mod_low_severity_pixels * pixel_area
mod_high_severity_area = mod_high_severity_pixels * pixel_area
high_severity_area = high_severity_pixels * pixel_area

# Print the area of each class or label
print('Enhanced regrowth,high area:', regrowth_high_area, 'hectares')
print('Enhanced regrowth, low area:', regrowth_low_area, 'hectares')
print('Unburned area:', unburned_area, 'hectares')
print('Low severity area:', low_severity_area, 'hectares')
print('Moderate-low severity area:', mod_low_severity_area, 'hectares')
print('Moderate-high severity area:', mod_high_severity_area, 'hectares')
print('High severity area:', high_severity_area, 'hectares')
Enhanced regrowth,high area: 0.004 hectares
Enhanced regrowth, low area: 0.2932 hectares
Unburned area: 74.45960000000001 hectares
Low severity area: 6.163600000000001 hectares
Moderate-low severity area: 0.15760000000000002 hectares
Moderate-high severity area: 0.0 hectares
High severity area: 0.0 hectares
```

TESTING WHETHER THE dNBR WAS CORRELATED TO dNDVI

```
# dNBR and dNDVI are 2D arrays with the same shape
dNBR_flat = dNBR.flatten()
dNDVI_flat = dNDVI.flatten()

# Calculate the linear regression line
slope, intercept, r_value, p_value, std_err = linregress(dNBR_flat, dNDVI_flat)
line = slope * dNBR_flat + intercept

# Create a scatter plot
fig, ax = plt.subplots()
ax.scatter(dNBR_flat, dNDVI_flat, alpha=0.5)

# Add the linear regression line to the scatter plot with the equation and correlation coefficient
eqn_label = f'dNDVI = {slope:.2f}(dNBR) + {intercept:.2f}'
corr_label = f'r = {r_value:.2f}'
ax.plot(dNBR_flat, line, color='red', label=f'{eqn_label}\n{corr_label}')

# Set the x-axis and y-axis labels
ax.set_xlabel('dNBR')
ax.set_ylabel('dNDVI')

# Set the title of the plot
ax.set_title('Plot of dNDVI versus dNBR')

# Add the legend to the plot
ax.legend()

plt.show()
```




```
'''
--------------------------------------------------------------
The following block of code originates and is modified from:

Balzter, H. (2023)
Materials for GY7709 masters computer classes.
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-
top%2FGY7709%5FSatellite%5FData%5FAnalysis%
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper-
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1
Downloaded 1 February 2023
--------------------------------------------------------------
'''
#To save a notebook as an html file, i add a new code cell at the end of the notebook
with the following contents:
```

```
!pip install -U notebook-as-pdf
!pyppeteer-install
!jupyter nbconvert /content/drive/MyDrive/satelliteCW1/229010645_GY7709_CW2.ipynb --to html
```