

Article

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

Ivan Innocent Sekibenga¹

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].

Citation: To be added by editorial staff during production.

Academic Editor: Firstname Lastname

Received: date

Revised: date

Accepted: date

Published: date



Copyright: © 2023 by the authors.

Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

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

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.

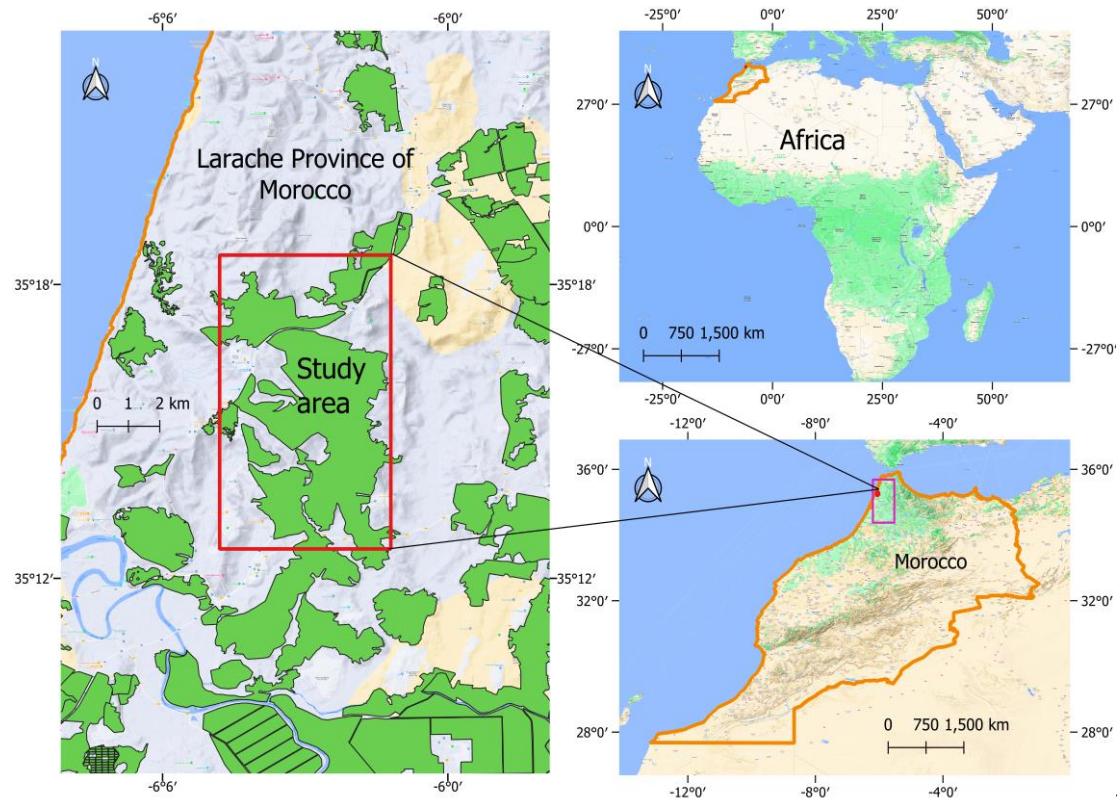
Early studies have used remote sensing techniques to detect and monitor wildfires and their effects on ecosystems [9,10]. Some researchers used moderate-resolution imaging spectroradiometer (MODIS) and Landsat imagery for wildfire detection [14,15], while others have investigated the potential of high-resolution 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)

78



(b)

79

80

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).

81

82

83

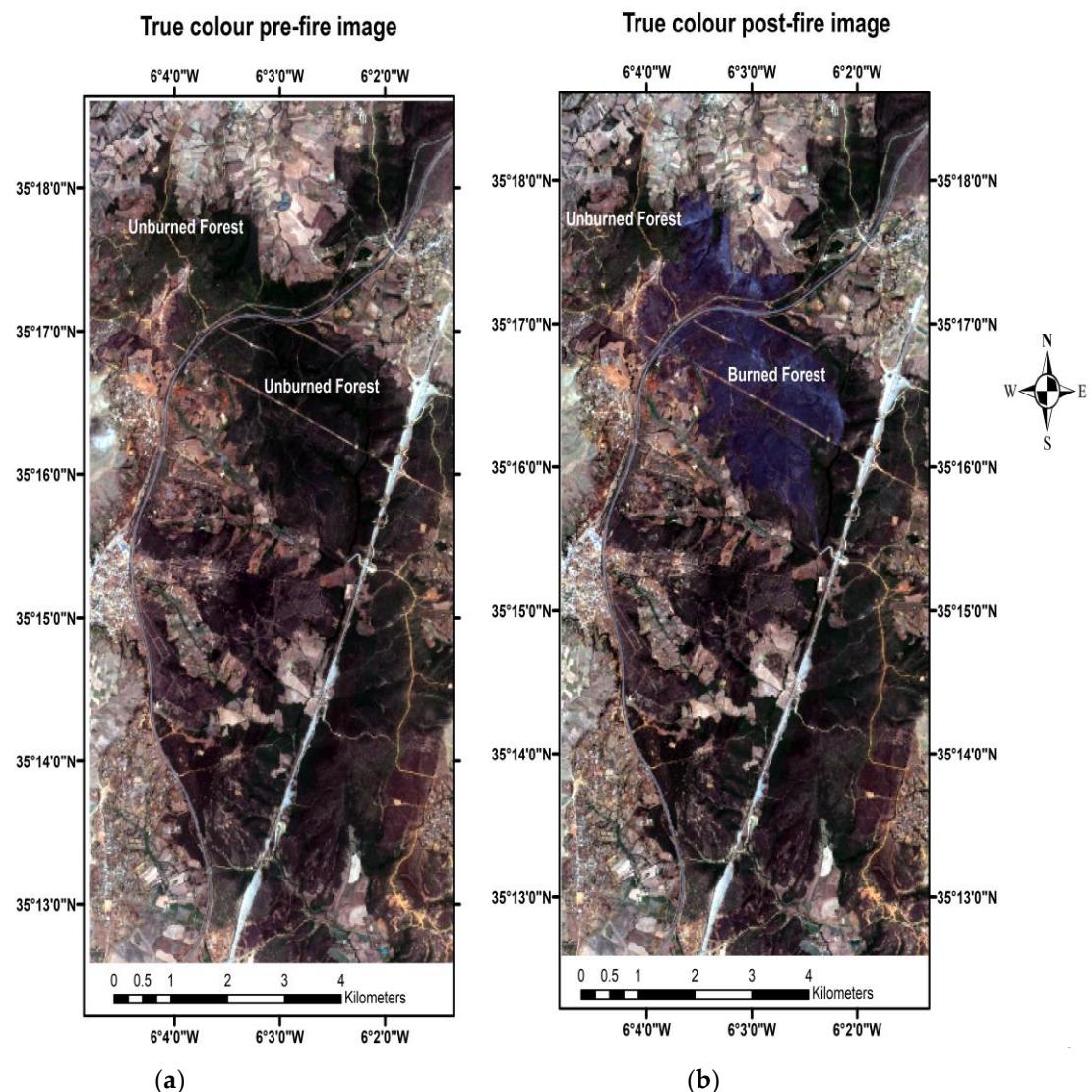


Figure 2. (a) True colour pre-fire image of Larache study area showing areas with unburned forests; (b) True colour post-fire image of Larache study area showing both burned and unburned forest areas.

85
86
87
88

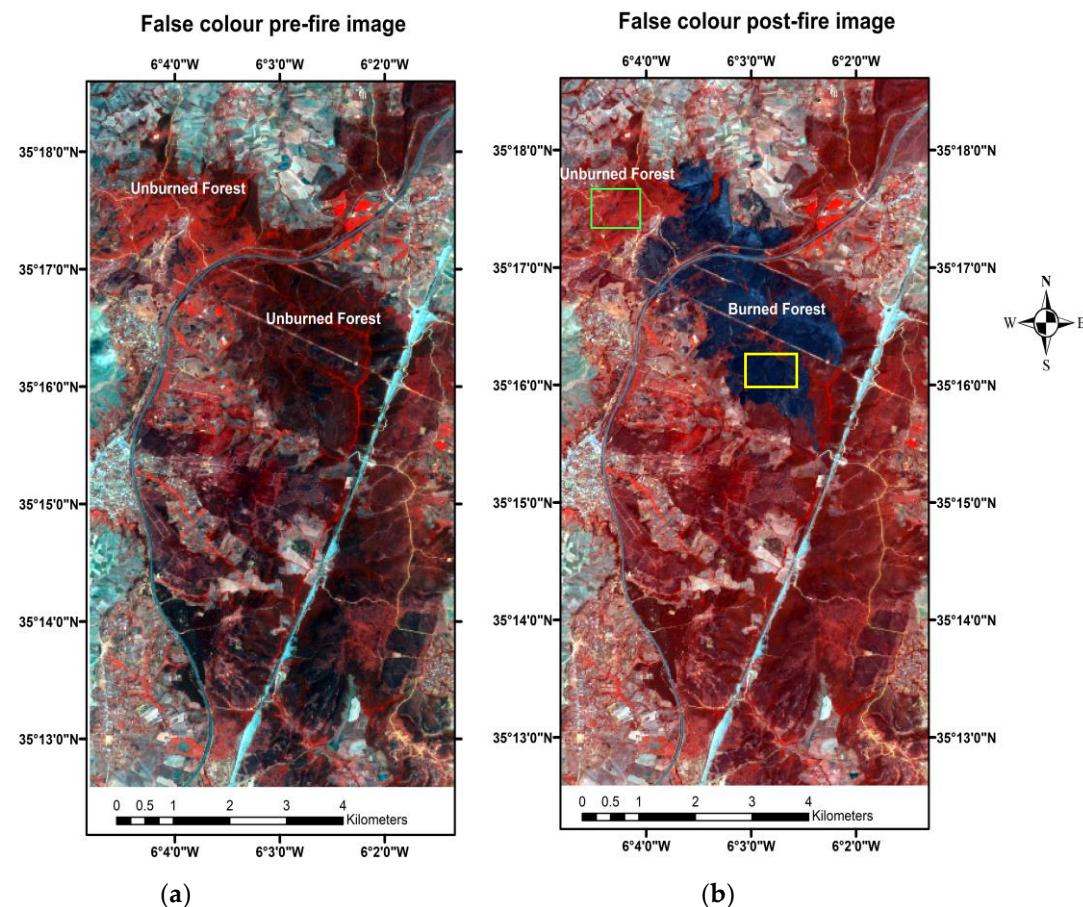


Figure 3. (a) False colour pre-fire image of Larache study area showing areas with unburned forests; (b) False colour post-fire image of Larache study area showing areas with burned and unburned forest polygons

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].

$$\text{NBR} = \frac{\text{NIR} - \text{SWIR}}{\text{NIR} + \text{SWIR}} = \frac{\text{B8} - \text{B12}}{\text{B8} + \text{B12}}$$
(1) 131

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]

$$\text{dNBR} = \text{Pre-fire NBR} - \text{Post-fire NBR}$$
(2) 140

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.

$$\text{NDVI} = \frac{\text{NIR} - \text{R}}{\text{NIR} + \text{R}} = \frac{\text{B8} - \text{B4}}{\text{B8} + \text{B4}}$$

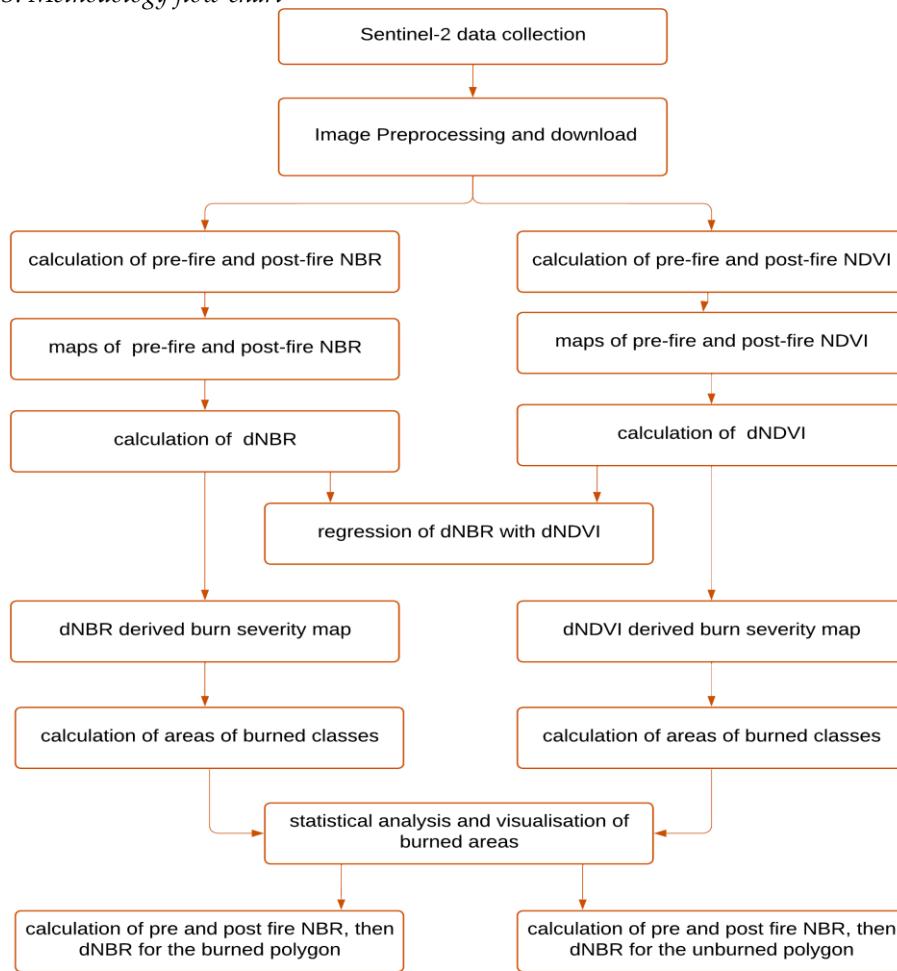
The differenced Normalised Differenced Vegetation Index(dNDVI) is obtained by taking the difference between the NDVI of the pre-fire image and the NDVI of the post-fire image of a study area. It is a quantifier of changes in NDVI values over a period of time [17,20].

$$\text{dNDVI} = \text{Pre-fire NDVI} - \text{Post-fire NDVI}$$

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

167



168

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

169

3. Results

170

3.1. Normalised Burn Ratios for Larache

171

172

3.1.1. Normalised Burn Ratio and differenced Normalised Burn Ratio values

173

174

	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

175

176

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.

177

178

179

180

3.1.2. Maps of Normalised Burn Ratio before and after fire

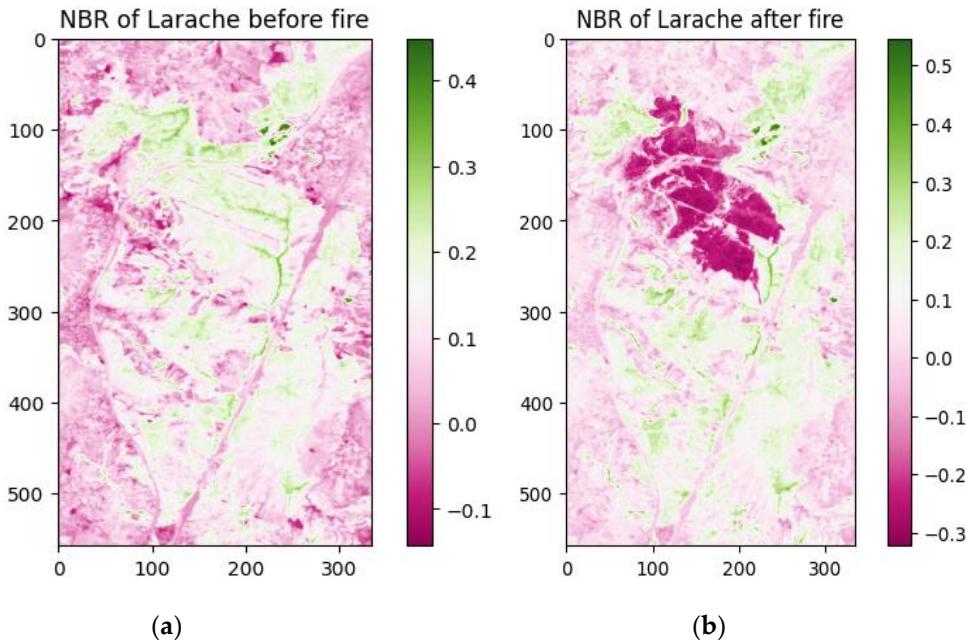


Figure 5. Green colour represents areas with healthy, green vegetation while Purple colour represents areas where vegetation has been burned or removed (a) Pre-fire Normalised Burn Ratio map of Larache study area; (b) Post-fire Normalised Burn Ratio map of Larache study area.

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

212

213

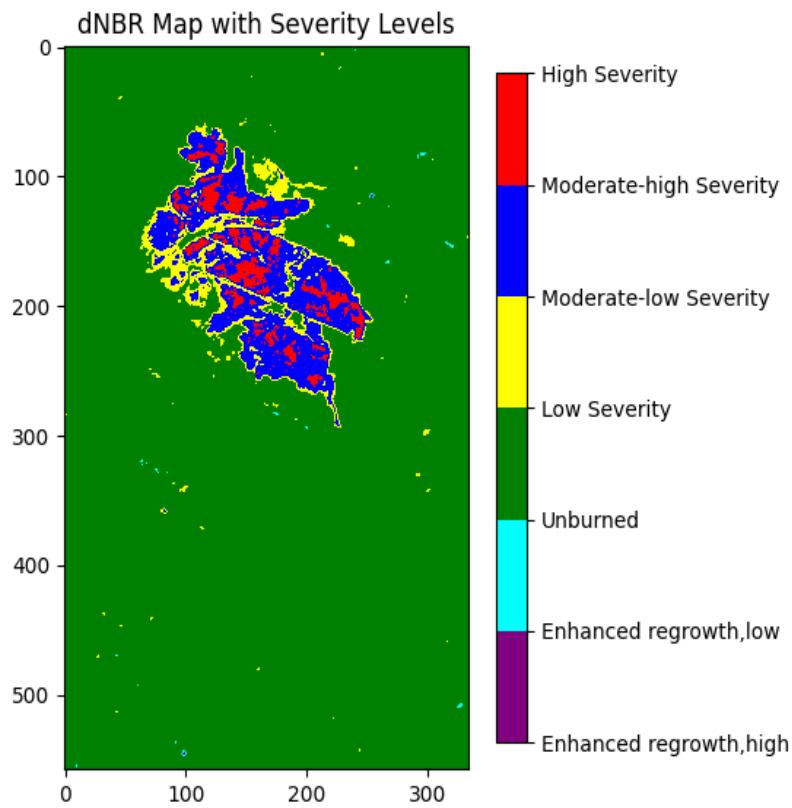


Figure 6. Differenced Normalised Burn Ratio map of Larache study area showing burn severity levels. Red represents areas of moderate-high burn severity to high severity, Blue represents areas of moderate-low to moderate-high burn severity, Yellow represents areas of low to moderate-low burn severity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

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

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

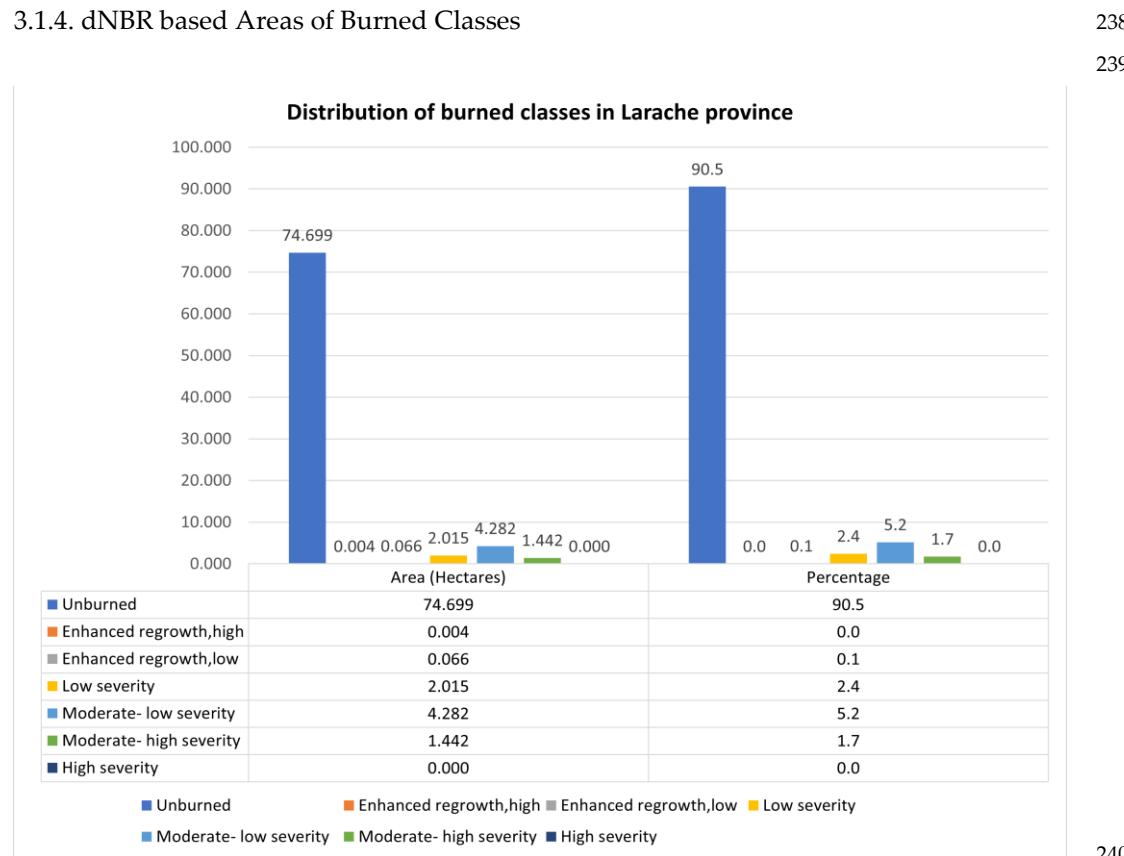


Figure 7. Distribution of Burn severity classes derived from dNBR index with their areas in hectares.

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 Differentiated Vegetation Indices for Larache

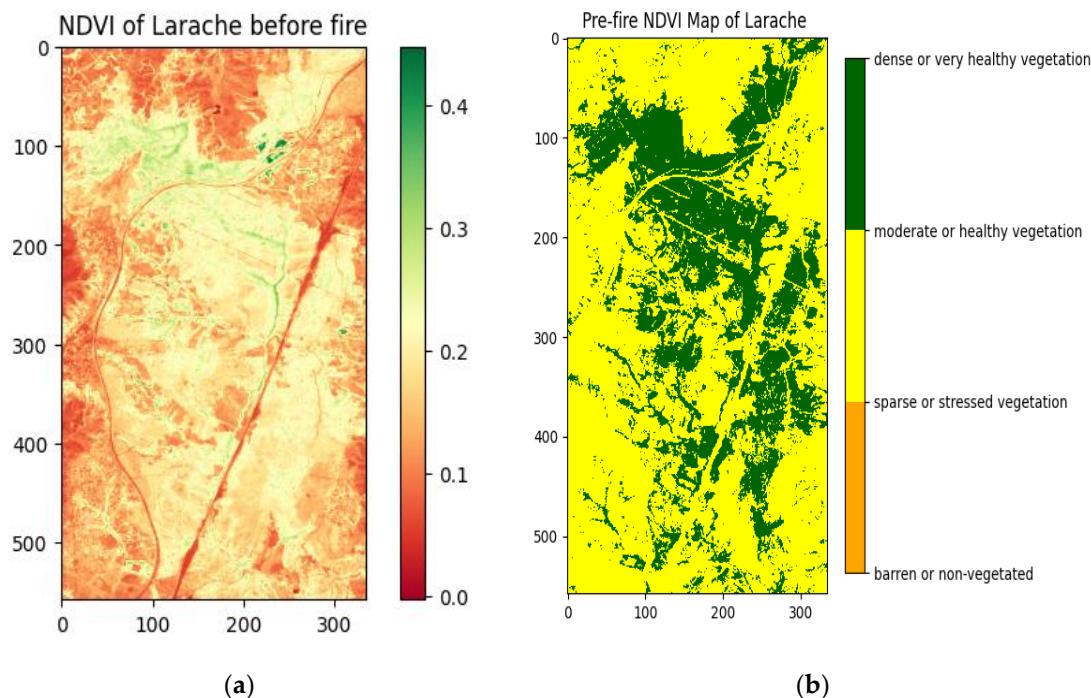
3.2.1. Summary of Normalised Differentiated Vegetation Indices for Larache

	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

260
261

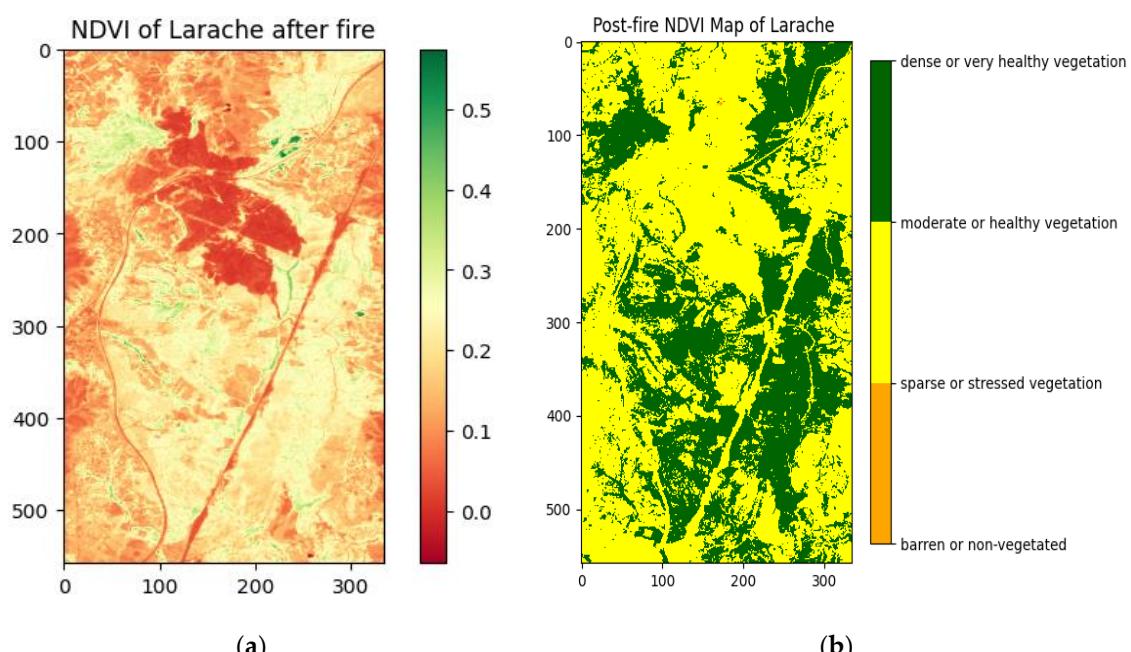
(a)

(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.

262
263
264
265
266
267
268

3.2.3. Mapping post- fire NDVI of Larache

269
270

(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;

271
272

(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.

273
274
275

276

277

3.2.4. Map of differenced Normalised Differentiated Vegetation Index for Larache

278

279

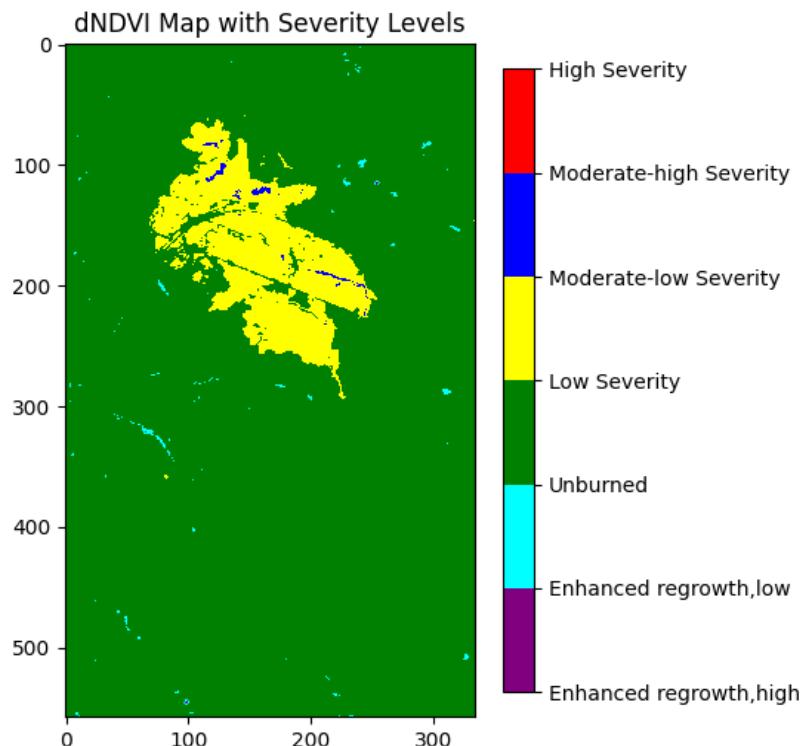


Figure 10. Differenced Normalised Difference Vegetation Index based map for Larache. Blue represents moderate-low burn severity areas, Yellow represents low-severity burn areas, Green represents unburned areas and cyan represents enhanced regrowth areas.

280
281
282
283

284

285

286

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**).

287
288
289
290
291
292
293

294

295

296

297

298

299

3.2.5. dNDVI derived Areas of Burned Classes

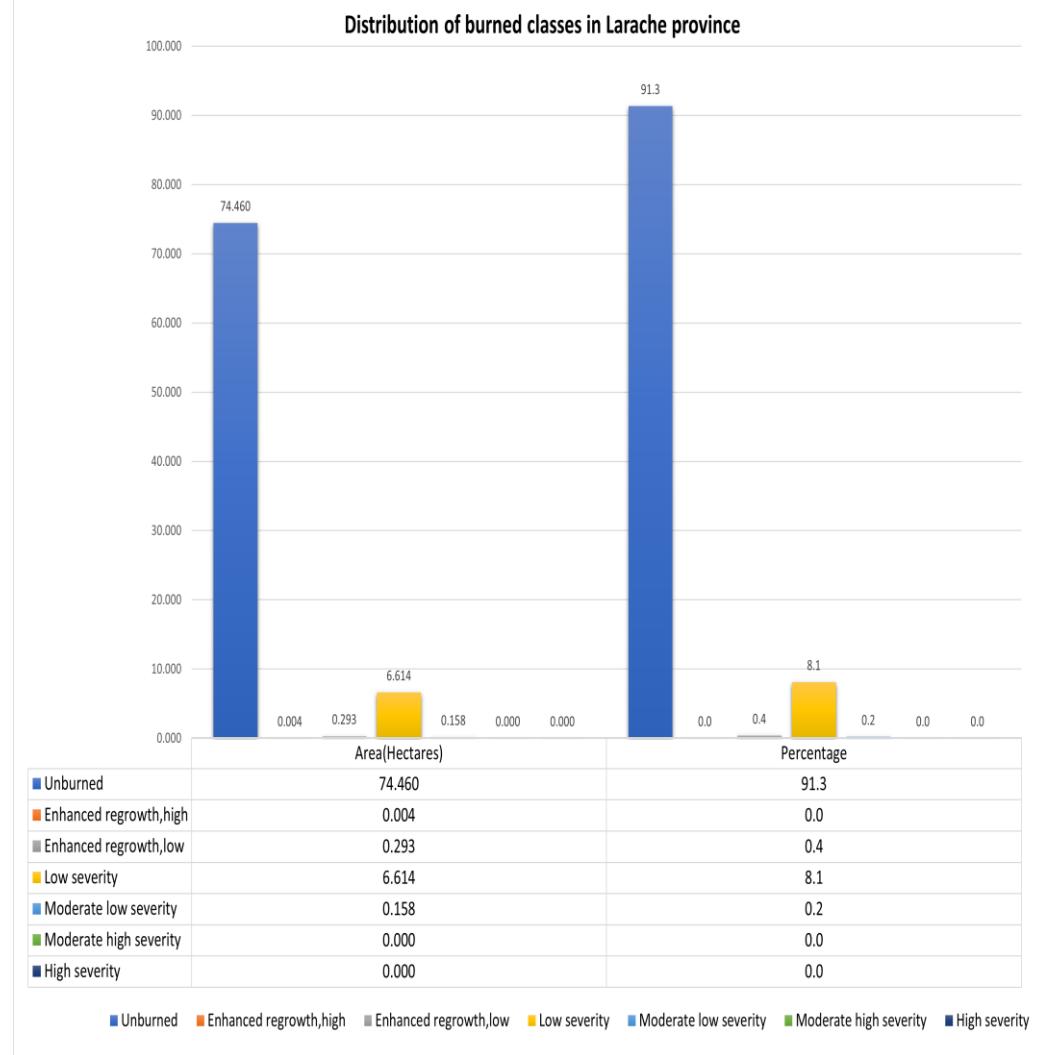


Figure 11. Distribution of Burn severity classes derived using dNDVI index with their areas in hectares.

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.

305
306
307
308
309
310
311
312
313
314
315
316
317
318

319
320
321

3.3 Relationship between dNDVI and dNBR

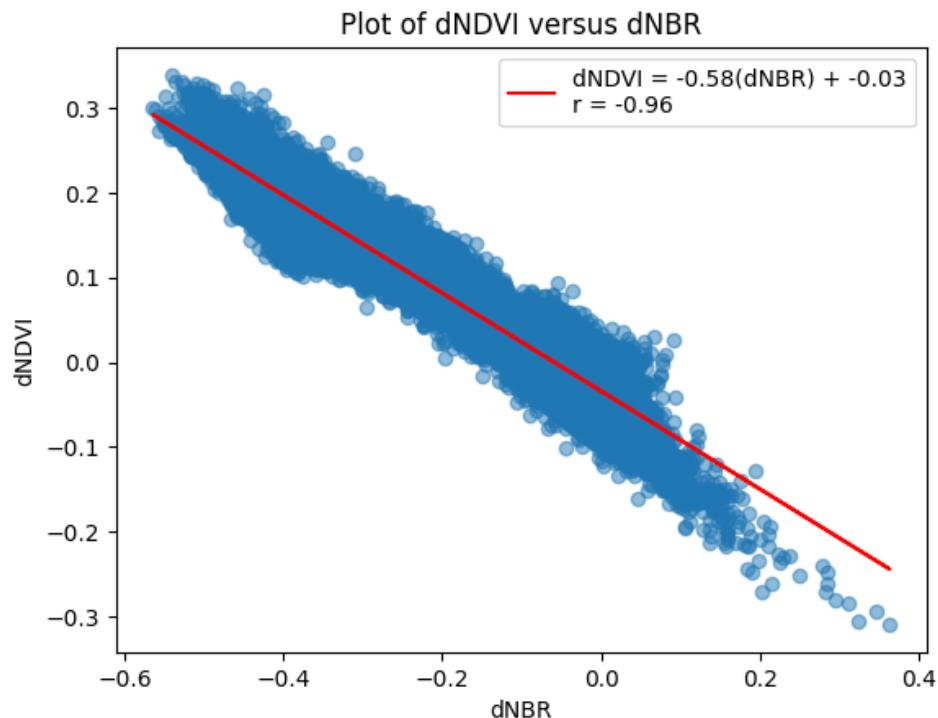


Figure 12. Plot of differenced Normalized Difference Vegetation Index against differenced Normalised Burn Ratio for Larache.

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.

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

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.

322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349

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 0.0868 (**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.	392
Conflicts of Interest: The author declares no conflict of interest.	395
References	396
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. <i>J. Biogeogr.</i> 2011 , <i>38</i> (12), 2223–2236. DOI: 10.1111/j.1365-2699.2011.02595.x	397
2. Westerling, A. L. Increasing Western US Forest Wildfire Activity: Sensitivity to Changes in the Timing of Spring. <i>Philos. Trans. R. Soc. B Biol. Sci.</i> 2016 , <i>371</i> (1696), 20150178. DOI: 10.1098/rstb.2015.0178	401
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. <i>For. Ecol. Manag.</i> 2013 , <i>294</i> , 54–61. DOI: 10.1016/j.foreco.2012.10.022	404
4. Lionello, P.; Scarascia, L. The Relation between Climate Change in the Mediterranean Region and Global Warming. <i>Reg. Environ. Change</i> 2018 , <i>18</i> (5), 1481–1493. DOI: 10.1007/s10113-018-1290-1	407
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. <i>Ecol. Model.</i> 2010 , <i>221</i> (1), 46–58. DOI: 10.1016/j.ecolmodel.2008.11.017	411
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. <i>Remote Sens. Environ.</i> 2008 , <i>112</i> (9), 3690–3707. DOI: 10.1016/j.rse.2008.05.013	415
7. Pflugmacher, D.; Cohen, W. B.; Kennedy, R. E. Using Landsat-Derived Disturbance History (1972–2010) to Predict Current Forest Structure. <i>Remote Sens. Environ.</i> 2012 , <i>122</i> , 146–165. DOI: 10.1016/j.rse.2011.09.025	418
8. Fuerst-Bjelis, B. The Fire in the Mediterranean Region: A Case Study of Forest Fires in Portugal. In <i>Mediterranean Identities - Environment, Society, Culture</i> ; 2017 ; Chapter 13. DOI: 10.5772/intechopen.69410	421
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.	424
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). <i>Remote Sens. Environ.</i> 2007 , <i>109</i> (1), 66–80. DOI: 10.1016/j.rse.2006.12.006	428
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. <i>Remote Sens. Environ.</i> 2012 , <i>120</i> , 25–36. DOI: 10.1016/j.rse.2011.11.026	430
12. Gorelick, N.; Hancher, M.; Dixon, M.; Ilyushchenko, S.; Thau, D.; Moore, R. Google Earth Engine: Planetary-Scale Geospatial Analysis for Everyone. <i>Remote Sens. Environ.</i> 2017 , <i>202</i> , 18–27. DOI: 10.1016/j.rse.2017.06.031	434
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.	437
14. López-García, M. J.; Caselles, V. Mapping Burns and Natural Reforestation Using Thematic Mapper Data. <i>Geocarto Int.</i> 1991 , <i>6</i> (1), 31–37. DOI: 10.1080/10106049109354290	441
15. Epting, J.; Verbyla, D.; Sorbel, B. Evaluation of Remotely Sensed Indices for Assessing Burn Severity in Interior Alaska Using Landsat TM and ETM+. <i>Remote Sens. Environ.</i> 2005 , <i>96</i> (3-4), 328–339. DOI: 10.1016/j.rse.2005.03.002	446

-
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 449
450
451
17. Soverel, N. O.; Perrakis, D. D.; Coops, N. C. Estimating Burn Severity from Landsat Differentiated 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.2010.03.013 452
453
454
455
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.2010.12.005 456
457
458
459
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 460
461
462
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 464
465
466

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

467
468
469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

Appendix: Google Colab Python Notebook	488
	489
Mapping fire damage from the wildfires in Larache Province, Morocco, in 2022 using Sentinel-2 Normalised Burn Ratio (NBR) and differenced NBR1	490
	491
Connecting to Google Drive from Colab.	492
# Loading the Drive helper and mount your Google Drive as a drive in the virtual machine	493
	494
from google.colab import drive	495
drive.mount('/content/drive')	496
Mounted at /content/drive	497
Importing all required libraries	498
# Installing some libraries that are not on Colab by default	499
!pip install rasterio	500
!pip install geopandas	501
!pip install rasterstats	502
!pip install earthengine-api	503
!pip install requests	504
!pip install sentinel-sat	505
	506
# Importing libraries	507
import geopandas as gpd	508
import rasterio	509
from rasterio import plot	510
from rasterio.plot import show_hist	511
import matplotlib.pyplot as plt	512
import numpy as np	513
from osgeo import gdal, ogr, osr	514
import json	515
import os	516
from os import listdir	517
from os.path import isfile, isdir, join	518
import math	519
from pprint import pprint	520
import shutil	521
import sys	522
import zipfile	523
import requests	524

```
import io 525
import webbrowser 526
import ee 527
import pandas as pd 528
import rasterio.mask 529
import xarray as xr 530
import matplotlib.colors as colors 531
import matplotlib 532
import glob 533
from rasterio.plot import show 534
from rasterio.mask import mask 535
from shapely.geometry import mapping 536
from scipy.stats import linregress 537
''' 538
-----
The following block of code originates and is modified from: 541
542
Balzter, H. (2023) 543
Materials for GY7709 masters computer classes. 544
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15 545
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk- 546
top%2FGY7709%5FSatellite%5FData%5FAnalysis% 547
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper- 548
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop% 549
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1 550
Downloaded 1 February 2023 551
-----
''' 552
# defining the root directory where our data are to be stored 553
rootdir = '/content/drive/MyDrive/satelliteCW1' # this is where pygge.py is saved on my 554
Google Drive 556
557
if rootdir not in sys.path: 558
    sys.path.append(rootdir) 559
# importing the pygge library of functions for this module 560
import pygge 561
562
%matplotlib inline 563
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/pub- 564
lic/simple/ 565
Collecting rasterio 566
```

```
Downloading rasterio-1.3.6-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl  
(20.0 MB) 567  
20.0/20.0 MB 5 568  
7.0 MB/s eta 0:00:00 569  
Collecting affine (from rasterio) 570  
    Downloading affine-2.4.0-py3-none-any.whl (15 kB) 571  
Requirement already satisfied: attrs in /usr/local/lib/python3.10/dist-packages (from rasterio) (23.1.0) 572  
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from rasterio) (2022.12.7) 573  
Requirement already satisfied: click>=4.0 in /usr/local/lib/python3.10/dist-packages (from rasterio) (8.1.3) 574  
Collecting cligj>=0.5 (from rasterio) 575  
    Downloading cligj-0.7.2-py3-none-any.whl (7.1 kB) 576  
Requirement already satisfied: numpy>=1.18 in /usr/local/lib/python3.10/dist-packages (from rasterio) (1.22.4) 577  
Collecting snuggs>=1.4.1 (from rasterio) 578  
    Downloading snuggs-1.4.7-py3-none-any.whl (5.4 kB) 579  
Collecting click-plugins (from rasterio) 580  
    Downloading click_plugins-1.1.1-py2.py3-none-any.whl (7.5 kB) 581  
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from rasterio) (67.7.2) 582  
Requirement already satisfied: pyparsing>=2.1.6 in /usr/local/lib/python3.10/dist-packages (from snuggs>=1.4.1->rasterio) (3.0.9) 583  
Installing collected packages: snuggs, cligj, click-plugins, affine, rasterio 584  
Successfully installed affine-2.4.0 click-plugins-1.1.1 cligj-0.7.2 rasterio-1.3.6 snuggs-1.4.7 585  
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/ 586  
Collecting geopandas 587  
    Downloading geopandas-0.13.0-py3-none-any.whl (1.1 MB) 588  
1.1/1.1 MB 14. 589  
6 MB/s eta 0:00:00 590  
Collecting fiona>=1.8.19 (from geopandas) 591  
    Downloading Fiona-1.9.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1 592  
6.0 MB) 593  
16.0/16.0 MB 9 594  
.2 MB/s eta 0:00:00 595  
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from geopandas) (23.1) 596  
Requirement already satisfied: pandas>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from geopandas) (1.5.3) 597  
608
```

```
Collecting pyproj>=3.0.1 (from geopandas) 609
  Downloading pyproj-3.5.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (610
  7.7 MB) 611
  └── 7.7/7.7 MB 38. 612
    8 MB/s eta 0:00:00 613
Requirement already satisfied: shapely>=1.7.1 in /usr/local/lib/python3.10/dist-packages 614
  (from geopandas) (2.0.1) 615
Requirement already satisfied: attrs>=19.2.0 in /usr/local/lib/python3.10/dist-packages 616
  (from fiona>=1.8.19->geopandas) (23.1.0) 617
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from 618
  fiona>=1.8.19->geopandas) (2022.12.7) 619
Requirement already satisfied: click~8.0 in /usr/local/lib/python3.10/dist-packages (f 620
  rom fiona>=1.8.19->geopandas) (8.1.3) 621
Requirement already satisfied: click-plugins>=1.0 in /usr/local/lib/python3.10/dist-pac 622
  kages (from fiona>=1.8.19->geopandas) (1.1.1) 623
Requirement already satisfied: cligj>=0.5 in /usr/local/lib/python3.10/dist-packages (f 624
  rom fiona>=1.8.19->geopandas) (0.7.2) 625
Collecting munch>=2.3.2 (from fiona>=1.8.19->geopandas) 626
  Downloading munch-2.5.0-py2.py3-none-any.whl (10 kB) 627
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist 628
  -packages (from pandas>=1.1.0->geopandas) (2.8.2) 629
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages 630
  (from pandas>=1.1.0->geopandas) (2022.7.1) 631
Requirement already satisfied: numpy>=1.21.0 in /usr/local/lib/python3.10/dist-packages 632
  (from pandas>=1.1.0->geopandas) (1.22.4) 633
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from mun 634
  ch>=2.3.2->fiona>=1.8.19->geopandas) (1.16.0) 635
Installing collected packages: pyproj, munch, fiona, geopandas 636
Successfully installed fiona-1.9.3 geopandas-0.13.0 munch-2.5.0 pyproj-3.5.0 637
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/pub 638
  lic/simple/ 639
Collecting rasterstats 640
  Downloading rasterstats-0.18.0-py3-none-any.whl (17 kB) 641
Requirement already satisfied: affine<3.0 in /usr/local/lib/python3.10/dist-packages (f 642
  rom rasterstats) (2.4.0) 643
Requirement already satisfied: click>7.1 in /usr/local/lib/python3.10/dist-packages (fr 644
  om rasterstats) (8.1.3) 645
Requirement already satisfied: cligj>=0.4 in /usr/local/lib/python3.10/dist-packages (f 646
  rom rasterstats) (0.7.2) 647
Collecting fiona<1.9 (from rasterstats) 648
  Downloading Fiona-1.8.22-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (649
  16.6 MB) 650
```

2.1 MB/s eta 0:00:00 16.6/16.6 MB 1 651
Requirement already satisfied: numpy>=1.9 in /usr/local/lib/python3.10/dist-packages (from rasterstats) (1.22.4) 652
Requirement already satisfied: rasterio>=1.0 in /usr/local/lib/python3.10/dist-packages (from rasterstats) (1.3.6) 653
Requirement already satisfied: rasterstats 654
Collecting simplejson (from rasterstats) 655
 Downloading simplejson-3.19.1-cp310-cp310-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl (137 kB) 656
137.9/137.9 kB 15 659
.9 MB/s eta 0:00:00 660
Requirement already satisfied: shapely in /usr/local/lib/python3.10/dist-packages (from rasterstats) (2.0.1) 661
Requirement already satisfied: attrs>=17 in /usr/local/lib/python3.10/dist-packages (from fiona<1.9->rasterstats) (23.1.0) 662
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from fiona<1.9->rasterstats) (2022.12.7) 663
Requirement already satisfied: click-plugins>=1.0 in /usr/local/lib/python3.10/dist-packages (from fiona<1.9->rasterstats) (1.1.1) 664
Requirement already satisfied: six>=1.7 in /usr/local/lib/python3.10/dist-packages (from fiona<1.9->rasterstats) (1.16.0) 665
Requirement already satisfied: munch in /usr/local/lib/python3.10/dist-packages (from fiona<1.9->rasterstats) (2.5.0) 666
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from fiona<1.9->rasterstats) (67.7.2) 667
Requirement already satisfied: snuggs>=1.4.1 in /usr/local/lib/python3.10/dist-packages (from rasterio>=1.0->rasterstats) (1.4.7) 668
Requirement already satisfied: pyparsing>=2.1.6 in /usr/local/lib/python3.10/dist-packages (from snuggs>=1.4.1->rasterio>=1.0->rasterstats) (3.0.9) 669
Installing collected packages: simplejson, fiona, rasterstats 670
Attempting uninstall: fiona 671
 Found existing installation: Fiona 1.9.3 672
 Uninstalling Fiona-1.9.3: 673
 Successfully uninstalled Fiona-1.9.3 674
Successfully installed fiona-1.8.22 rasterstats-0.18.0 simplejson-3.19.1 675
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/ 676
Requirement already satisfied: earthengine-api in /usr/local/lib/python3.10/dist-packages (0.1.350) 677
Requirement already satisfied: google-cloud-storage in /usr/local/lib/python3.10/dist-packages (from earthengine-api) (2.8.0) 678
680
681
682
683
684
685
686
687
688
689
690
691

Requirement already satisfied: google-api-python-client>=1.12.1 in /usr/local/lib/python3.10/dist-packages (from earthengine-api) (2.84.0)	692
Requirement already satisfied: google-auth>=1.4.1 in /usr/local/lib/python3.10/dist-packages (from earthengine-api) (2.17.3)	694
Requirement already satisfied: google-auth-httplib2>=0.0.3 in /usr/local/lib/python3.10/dist-packages (from earthengine-api) (0.1.0)	696
Requirement already satisfied: httplib2<1dev,>=0.9.2 in /usr/local/lib/python3.10/dist-packages (from earthengine-api) (0.21.0)	698
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from earthengine-api) (2.27.1)	700
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)	702
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)	705
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)	707
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)	709
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)	711
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)	713
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)	715
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)	718
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)	720
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->earthengine-api) (1.26.15)	722
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->earthengine-api) (2022.12.7)	724
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.10/dist-packages (from requests->earthengine-api) (2.0.12)	726
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->earthengine-api) (3.4)	729
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)	732

```
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 ( 733  
from google-api-core!=2.0.*,!2.1.*,!2.2.*,!2.3.0,<3.0.0dev,>=1.31.5->google-api-pyth 734  
on-client>=1.12.1->earthengine-api) (3.20.3) 735  
Requirement already satisfied: google-crc32c<2.0dev,>=1.0 in /usr/local/lib/python3.10/ 736  
dist-packages (from google-resumable-media>=2.3.2->google-cloud-storage->earthengine-ap 737  
i) (1.5.0) 738  
Requirement already satisfied: pyasn1<0.6.0,>=0.4.6 in /usr/local/lib/python3.10/dist-p 739  
ackages (from pyasn1-modules>=0.2.1->google-auth>=1.4.1->earthengine-api) (0.5.0) 740  
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/pub 741  
lic/simple/ 742  
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (2.2 743  
7.1) 744  
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.10/dist- 745  
packages (from requests) (1.26.15) 746  
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-pac 747  
kages (from requests) (2022.12.7) 748  
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.10/d 749  
ist-packages (from requests) (2.0.12) 750  
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages 751  
(from requests) (3.4) 752  
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/pub 753  
lic/simple/ 754  
Collecting sentinelst 755  
Downloaded sentinelst-1.2.1-py3-none-any.whl (48 kB) 756  
----- 48.8/48.8 kB 3 757  
.0 MB/s eta 0:00:00 758  
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (fro 759  
m sentinelst) (2.27.1) 760  
Requirement already satisfied: click>=7.1 in /usr/local/lib/python3.10/dist-packages (f 761  
rom sentinelst) (8.1.3) 762  
Collecting html2text (from sentinelst) 763  
Downloaded html2text-2020.1.16-py3-none-any.whl (32 kB) 764  
Collecting geojson>=2 (from sentinelst) 765  
Downloaded geojson-3.0.1-py3-none-any.whl (15 kB) 766  
Requirement already satisfied: tqdm>=4.58 in /usr/local/lib/python3.10/dist-packages (f 767  
rom sentinelst) (4.65.0) 768  
Collecting geomet (from sentinelst) 769  
Downloaded geomet-1.0.0-py3-none-any.whl (28 kB) 770  
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from geo 771  
met->sentinelst) (1.16.0) 772
```

Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->sentinelsat) (1.26.15)	774
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->sentinelsat) (2022.12.7)	776
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.10/dist-packages (from requests->sentinelsat) (2.0.12)	778
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->sentinelsat) (3.4)	780
Installing collected packages: html2text, geomet, geojson, sentinelsat	782
Successfully installed geojson-3.0.1 geomet-1.0.0 html2text-2020.1.16 sentinelsat-1.2.1	783

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 .	784
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----

```
# 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 manually  
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_id=twmhr4UDalOCouqwF2Y0MIQtfo9eyX1OZ2Dz3JAMYyA&tc=OoXrSQp4z0FUtOsxtSs8rJbXTb8wirKhX5pYjvFP6cc&cc=EuRBf2I1jjH17quMcxhdb_xqE6glxrecpELEwCgbrO4

The authorization workflow will generate a code, which you should paste in the box below.

Enter verification code: 4/1AbUR2VP2WYIYV49woH8vir8w4hTYSfdMaTGVws3E8xj-66vhqTKzgik16gM

Successfully saved authorization token.

'''

The following block of code originates from:

Balzter, H. (2023) 813
Materials for GY7709 masters computer classes. 814
[https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15/onederive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1](https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15/onederive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1) 815
Downloaded 1 February 2023 816
----- 817
... 818
set up your directories for the satellite data 819
Note that we do all the downloading and data analysis on the temporary drive 820
on Colab. We will copy the output directory to our Google Drive at the end. 821
Colab has more disk space (about 40 GB free space) than Google Drive (15 GB). 822
However, the data on the Colab disk space are NOT kept when you log out. 823
path to your Google Drive 824
print("Connected to data directory: " + rootdir) 825
path to your temporary drive on the Colab Virtual Machine. This will be removed 826
when the session ends. 827
cd = "/content/work" 828
directory for downloading the Sentinel-2 composites 829
Note that we are using the 'join' function imported from the os library here 830
It is an easy way of merging strings into a directory structure. 831
It is clever and chooses the / or \ depending on whether you are on Windows or Linux. 832
downloaddir = join(cd, 'download') # where we save the downloaded images 833
CAREFUL: This code removes the named directories and everything inside them to free 834
up space 835
Because the downloaddir is in your temporary drive, it should be empty at the 836
start of the session. 837
Note: shutil provides a lot of useful functions for file and directory management 838
try: 839
 shutil.rmtree(downloaddir) 840
except: 841
 print(downloaddir + " not found.") 842
create the new directories, unless they already exist 843
844
845
846
847
848
849
850
851
852
853
854

```
os.makedirs(cd, exist_ok=True) 855
os.makedirs(downloadaddir, exist_ok=True) 856
print("Connected to Colab temporary data directory: " + cd) 857
858
print("\nList of contents of " + rootdir) 859
for f in sorted(os.listdir(rootdir)): 860
    print(f)
861
Connected to data directory: /content/drive/MyDrive/satelliteCW1 862
/content/work/download not found. 863
Connected to Colab temporary data directory: /content/work 864
865
List of contents of /content/drive/MyDrive/satelliteCW1 866
.ipynb_checkpoints 867
229010645_GY7709_CW1_BACKUP2.ipynb 868
229010645_GY7709_CW1_BACKUP3.ipynb 869
229010645_GY7709_CW1.html 870
229010645_GY7709_CW1.ipynb 871
229010645_GY7709_CW1_backup.ipynb 872
229010645_GY7709_CW1_original.html 873
229010645_GY7709_CW2.ipynb 874
NBR_larache_after.tif 875
NBR_larache_before.tif 876
NDVI_larache_after.tif 877
NDVI_larache_before.tif 878
__pycache__ 879
backup 880
burned_layers 881
dNBR.tif 882
dNBR.tiff 883
dNBR_warped.tif 884
larache_after 885
larache_before 886
larache_before_fire.tif 887
larache_before_fire_warped.tif 888
larache_new_shapefile 889
larache_shapefiles 890
leicestershire 891
oakham 892
practical_week32_SentinelSat.ipynb 893
pygge.ipynb 894
pygge.py 895
```

sencredentials.txt	897
taza	898
unburned_layers	899

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". 900

''' 902
----- 903

The following block of code is modified from: 905

Balzter, H. (2023) 907

Materials for GY7709 masters computer classes. 908

[https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1](https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1) 909

Downloaded 1 February 2023 915

----- 916

''' 917

EDITING THE SEARCH OPTIONS BELOW 918

shapefile = join(rootdir, 'larache_new_shapefile', 'FULL_POLYGON.shp') # ESRI Shapefile 920
of the study area 921

----- 922

checking whether the shapefile exists 923

if os.path.exists(shapefile): 924

 print('Shapefile found: '+shapefile) 925

else: 926

 print('ERROR: Shapefile not found: '+shapefile) 927

 print('Upload a shapefile to your Google Drive directory: '+ rootdir) 928

----- 929

Defining a date range for the search 930

datefrom = '2022-07-01' # start date for imagery search 931

dateto = '2022-07-10' # end date for imagery search 932

time_range = [datefrom, dateto] # format as a list 933

----- 934

Defining which cloud cover percentage we accept in the images 935

```
clouds = 8 # maximum acceptable cloud cover in 936
Shapefile found: /content/drive/MyDrive/satelliteCW1/larache_new_shapefile/FULL_POLYGON 937
.shp 938
```

Getting some information about the shapefile.

```
''' 940
----- 941
----- 942
The following block of code originates from: 943
----- 944
Balzter, H. (2023) 945
Materials for GY7709 masters computer classes. 946
https://uniofleicester-my.sharepoint.com/personal/hb91\_leicester\_ac\_uk/\_layouts/15/onederive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-top%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1 947
Downloaded 1 February 2023 952
----- 953
''' 954
----- 955
# Getting the shapefile layer's extent, CRS and EPSG code 956
extent, outSpatialRef, epsg = pygge.get_shp_extent(shapefile) 957
print("Extent of the area of interest (shapefile):\n", extent) 958
print(type(extent)) 959
print("\nCoordinate referencing system (CRS) of the shapefile:\n", outSpatialRef) 960
print('EPSG code: ', epsg) 961
Extent of the area of interest (shapefile): 962
(-6.08, -6.02, 35.21, 35.31) 963
<class 'tuple'> 964
----- 965
Coordinate referencing system (CRS) of the shapefile: 966
GEOGCS["WGS 84", 967
    DATUM["WGS_1984", 968
        SPHEROID["WGS 84", 6378137, 298.257223563, 969
            AUTHORITY["EPSG", "7030"]], 970
            AUTHORITY["EPSG", "6326"]], 971
        PRIMEM["Greenwich", 0, 972
            AUTHORITY["EPSG", "8901"]], 973
        UNIT["degree", 0.0174532925199433, 974
            AUTHORITY["EPSG", "9122"]], 975
```

```
AXIS["Latitude",NORTH], 976
AXIS["Longitude",EAST], 977
AUTHORITY["EPSG","4326"]] 978
EPSG code: 4326 979
3. Getting the extent of the shapefile into a format that Google Earth Engine understands. 980
981
''' 981
----- 982
The following block of code originates from: 983
984
Balzter, H. (2023) 985
Materials for GY7709 masters computer classes. 986
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15 987
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk- 988
top%2FGY7709%5FSatellite%5FData%5FAnalysis% 989
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper- 990
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop% 991
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1 992
Downloaded 1 February 2023 993
----- 994
''' 995
# GEE needs a special format for defining an area of interest. 996
# It has to be a GeoJSON Polygon and the coordinates should be first defined in a list 997
and then converted using ee.Geometry. 998
999
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], extent[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({ 1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
```

```
"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) 1060
Materials for GY7709 masters computer classes. 1061
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15/onederive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-top%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1 1062
Downloaded 1 February 2023 1063
----- 1064
''' 1065
Obtaining download links for image composites from an image collection on Google Earth Engine 1066
Name of the Sentinel 2 image collection 1067
s2collection = ('COPERNICUS/S2') 1068
getting the median composite of Sentinel-2 images in the time range 1069
s2median = pygge.obtain_image_sentinel(s2collection, time_range, search_area, clouds) 1070
Downloading the followig bands as a list of strings namely; Blue, Green, Red, NIR and SWIR 1071
bands = ['B2', 'B3', 'B4', 'B8', 'B12'] 1072
print(bands) 1073
spatial resolution of the downloaded data 1074
resolution = 20 # in units of metres 1075
Download images in Geotiff, using the get_url(name, image, scale, region) method 1076
'region' is obtained from the area, but the format is adjusted using get_region(geom) 1077
method 1078
search_region = pygge.get_region(search_area) 1079
s2url = pygge.get_url('s2', s2median.select(bands), resolution, search_region, filePer-Band=False) 1080
print(s2url) 1081
['B2', 'B3', 'B4', 'B8', 'B12'] 1082
<https://earthengine.googleapis.com/v1alpha/projects/earthengine-legacy/thumbnails/44e3d97d9adfca34655516a4c2fea849-11e933b4d0f41ebc8d8b877d0589bca7:getPixels> 1083

Download the data for the before fire image of Larache

1100

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

1101

```
'''  
-----  
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/onederive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2Ezip&parent=%2Fpersonal%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(downloadaddir)  
  
# 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

1134

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

1135

1136

1137

1138

1139

Balzter, H. (2023) 1140
Materials for GY7709 masters computer classes. 1141
[https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15/onederive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDeskt... 1142](https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15/onederive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDeskt...)
/top%2FGY7709%5FSatellite%5FData%5FAnalysis% 1143
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop% 1144
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1 1145
Downloaded 1 February 2023 1146
----- 1147
''' 1148
where the downloaded Sentinel-2 images are stored 1149
os.chdir(downloadaddir) 1150
print("contents of ", downloadaddir, ":") 1151
!ls -l 1152
contents of /content/work/download : 1153
total 2120 1154
-rw-r--r-- 1 root root 2168674 May 12 13:28 s2.tif 1155
6. The downloaded file "s2.tif" is seen. 1156
7. the downloaded images are saved to a temporary directory that will be deleted when the virtual machine is 1157
closed. To save the images to my local directory, this is how it went; 1158
8. I went to my Google Colab folder in the panel on the left hand side. 1159
9. Found the download directory and clicked on a Sentinel-2 image folder. 1160
10. Right-clicked on it ,renamed it and selected 'download' to save it. 1161

Showing the before fire image of Larache as a colour composite

1164
''' 1165
----- 1166
The following block of code originates and is modified from: 1167
1168
Balzter, H. (2023) 1169
Materials for GY7709 masters computer classes. 1170
[https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15/onederive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDeskt... 1171](https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15/onederive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDeskt...)
/top%2FGY7709%5FSatellite%5FData%5FAnalysis% 1172
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop% 1173
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1 1174
----- 1175
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop% 1176

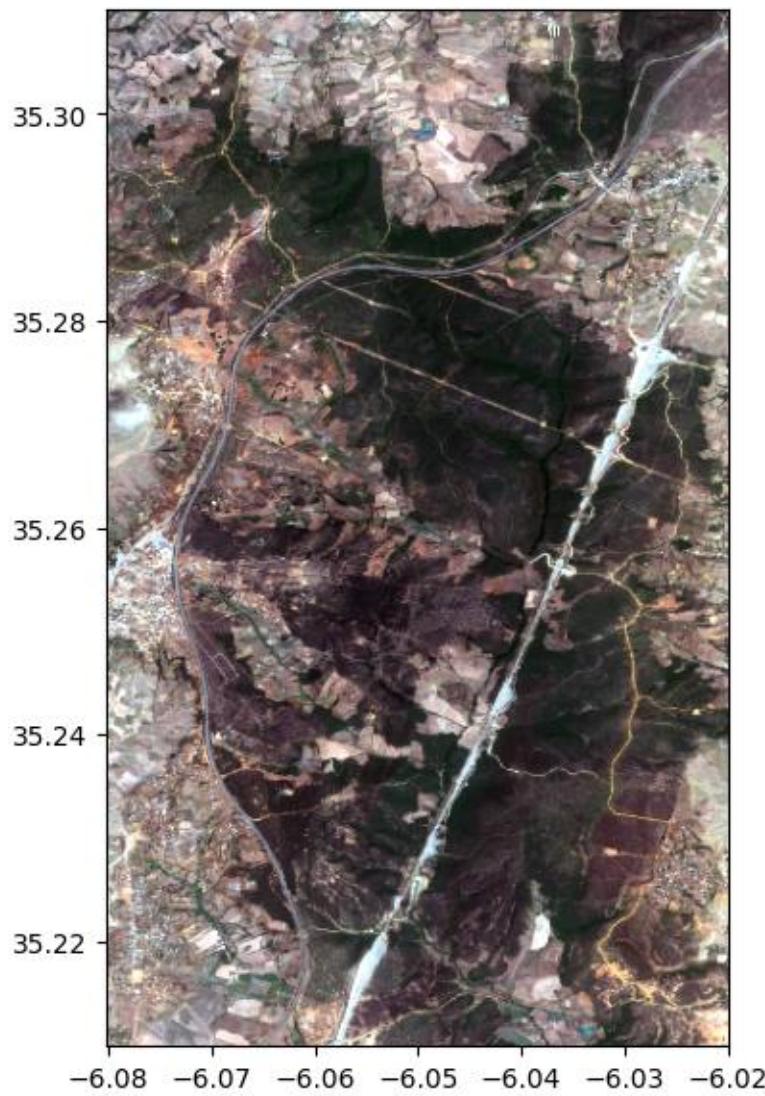
```
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1 1177  
Downloaded 1 February 2023 1178  
----- 1179  
''' 1180  
# getting list of all tiff files in the directory 1181  
allfiles = [f for f in listdir(downloaddir) if isfile(join(downloaddir, f))] 1182  
print(allfiles) 1183  
 1184  
# selecting the file for visualisation 1185  
thisfile = allfiles[0] 1186  
print(thisfile) 1187  
['s2.tif'] 1188  
s2.tif 1189
```

True Colour Composite of before fire image of Larache

```
 1190  
''' 1191  
----- 1192  
The following block of code originates and is modified from: 1193  
 1194  
Balzter, H. (2023) 1195  
Materials for GY7709 masters computer classes. 1196  
https://uniofleicester-my.sharepoint.com/personal/hb91\_leicester\_ac\_uk/\_layouts/15/onederive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1 1197  
Downloaded 1 February 2023 1198  
----- 1199  
''' 1200  
# creating a figure 1201  
fig, (ax1) = plt.subplots(1,1, figsize=(7,7)) 1202  
fig.patch.set_facecolor('white') 1203  
 1204  
# the downloaded file is float32 data format 1205  
# for plotting, uint8 data format is needed 1206  
 1207  
# plotting the image with full extent 1208  
pygge.easy_plot(thisfile, ax=ax1, bands=[3,2,1], percentiles=[0,99]) 1209  
 1210  
 1211  
 1212  
 1213  
 1214  
 1215
```

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

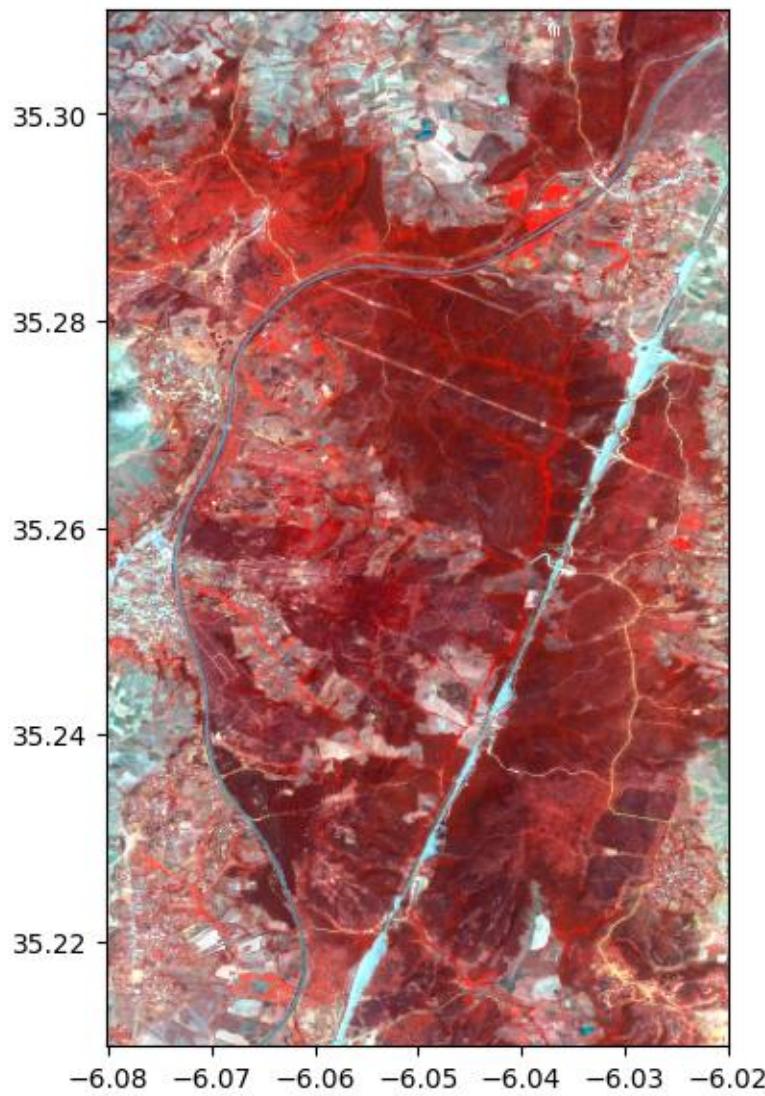


False Colour Composite of before fire image of Larache

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 1231  
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk- 1232  
top%2FGY7709%5FSatellite%5FData%5FAnalysis% 1233  
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper- 1234  
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop% 1235  
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1 1236  
Downloaded 1 February 2023 1237  
----- 1238  
''' 1239  
# creating a figure with 2x3 subplots 1240  
fig, (ax1) = plt.subplots(1,1, figsize=(7,7)) 1241  
fig.patch.set_facecolor('white') 1242  
1243  
# the downloaded file is float32 data format 1244  
# for plotting, uint8 data format is needed 1245  
1246  
# plotting the image with full extent 1247  
pygge.easy_plot(thisfile, ax=ax1, bands=[4,2,1], percentiles=[0,99]) 1248  
WARNING:rasterio._env:CPLE_AppDefined in s2.tif: TIFFReadDirectory:Sum of Photometric t 1249  
ype-related color channels and ExtraSamples doesn't match SamplesPerPixel. Defining non 1250  
-color channels as ExtraSamples. 1251  
WARNING:rasterio._env:CPLE_AppDefined in TIFFReadDirectory:Sum of Photometric type-rela 1252  
ted color channels and ExtraSamples doesn't match SamplesPerPixel. Defining non-color c 1253  
hannels as ExtraSamples. 1254
```



1255

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

1256

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

1257

1258

1259

The following block of code originates and is modified from:

1260

1261

1262

Balzter, H. (2023)

1263

Materials for GY7709 masters computer classes.

1264

https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15

1265

1266

1267

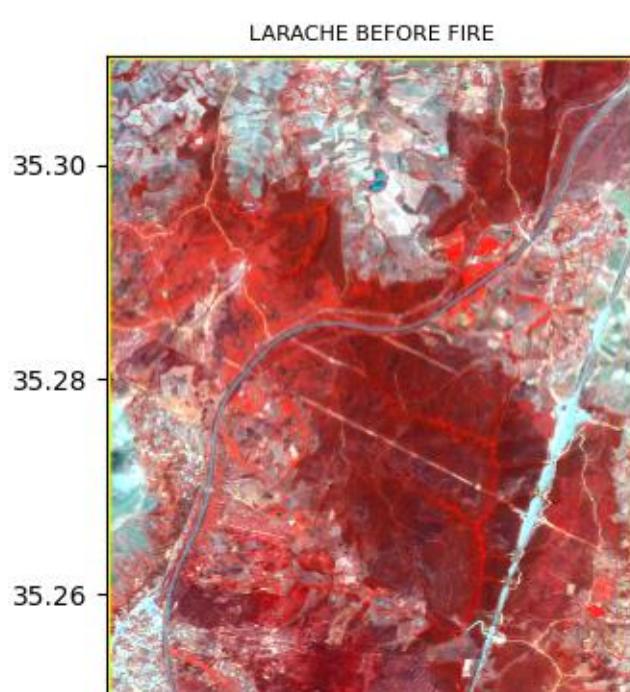
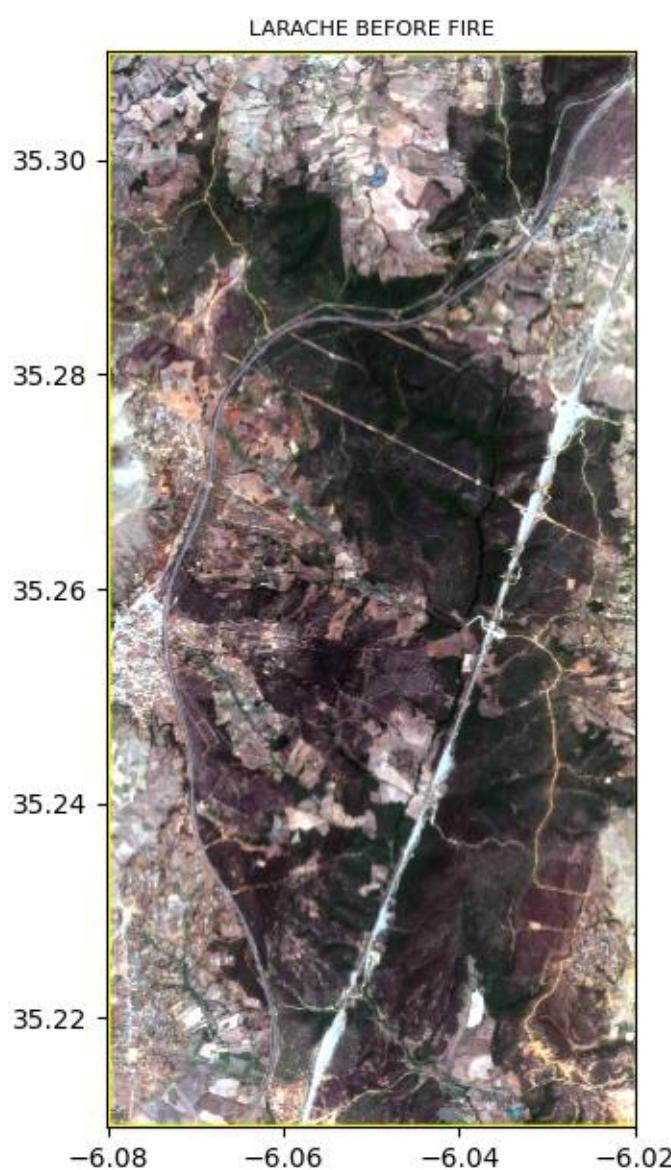
```
/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk- 1268  
top%2FGY7709%5FSatellite%5FData%5FAnalysis% 1269  
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper- 1270  
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop% 1271  
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1 1272  
Downloaded 1 February 2023 1273  
----- 1274  
''' 1275  
# printing the EPSG code of our shapefile into which we want to reproject the TCI images 1276  
print("Reprojecting image to EPSG projection ", epsg) 1277  
1278  
# making a file name for the new file 1279  
warpfile = thisfile.split(sep='.')[0] + '_warped.tif' 1280  
print("We are in this directory: ") 1281  
!pwd 1282  
print("Input file: ", thisfile) 1283  
print("Output file: ", warpfile) 1284  
1285  
# calling the easy_warp function 1286  
tmp = pygge.easy_warp(thisfile, warpfile, epsg) 1287  
Reprojecting image to EPSG projection 4326 1288  
We are in this directory: 1289  
/content/work/download 1290  
WARNING:rasterio._env:CPLE_AppDefined in s2.tif: TIFFReadDirectory:Sum of Photometric type-related color channels and ExtraSamples doesn't match SamplesPerPixel. Defining non-color channels as ExtraSamples. 1291  
WARNING:rasterio._env:CPLE_AppDefined in TIFFReadDirectory:Sum of Photometric type-related color channels and ExtraSamples doesn't match SamplesPerPixel. Defining non-color channels as ExtraSamples. 1292  
Input file: s2.tif 1293  
Output file: s2_warped.tif 1294  
Creating warped file:s2_warped.tif 1295  
1296  
1297  
1298  
1299  
1300
```

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. 1302
1303
''' 1304
----- 1305
The following block of code originates and is modified from: 1306

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%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1](https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fpersonal%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")
```



Defining the search parameters for the after fire image of Larache

1333

13. We change the search dates to get a new image after the fire had burnt out. Fires were first reported on 15 July 1334
2022 1335

''' 1336

----- 1337

The following block of code originates and is modified from: 1339

1340

Balzter, H. (2023) 1341

Materials for GY7709 masters computer classes. 1342

[https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1](https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1) 1343

1344

top%2FGY7709%5FSatellite%5FData%5FAnalysis% 1345

1346

5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper- 1347

1348

sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop% 1349

1350

2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1 1351

1352

Downloaded 1 February 2023 1353

1354

----- 1355

''' 1356

EDITing THE SEARCH OPTIONS BELOW 1357

1358

shapefile = join(rootdir, 'larache_shapefiles', 'FULL_POLYGON.shp') # ESRI Shapefile of 1359
the study area 1360

1361

checking whether the shapefile exists 1362

1363

if os.path.exists(shapefile): 1364

 print('Shapefile found: '+shapefile) 1365

else: 1366

 print('ERROR: Shapefile not found: '+shapefile) 1367

 print('Upload a shapefile to your Google Drive directory: '+ rootdir) 1368

1369

Defining a date range for the search 1370

1371

datefrom = '2022-07-30' # start date for imagery search 1372

1373

dateto = '2022-08-05' # end date for imagery search 1374

1375

time_range = [datefrom, dateto] # format as a list 1376

1377

Defining percentage cloud cover accepted in the images 1378

1379

clouds = 10 # maximum acceptable cloud cover in % 1380

Shapefile found: /content/drive/MyDrive/satelliteCW1/larache_shapefiles/FULL_POLYGON.sh 1381

1382

p

14. Get some information about our shapefile.

1373

' ''

1374

The following block of code originates and is modified from:

1375

Balzter, H. (2023)

1379

Materials for GY7709 masters computer classes.

1380

[https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15/onederive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1](https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15/onederive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1)

1381

Downloaded 1 February 2023

1387

' ''

1388

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'>

1389

Coordinate referencing system (CRS) of the shapefile:

1400

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

1401

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

1414

```
''' 1415
----- 1416
----- 1417
The following block of code originates and is modified from: 1418
1419
Balzter, H. (2023) 1420
Materials for GY7709 masters computer classes. 1421
https://uniофleicester-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%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1 1422
Downloaded 1 February 2023 1423
----- 1424
''' 1425
# GEE needs a special format for defining an area of interest. 1426
# It has to be a GeoJSON Polygon and the coordinates should be first defined in a list 1427
and then converted using ee.Geometry. 1428
extent_list = list(extent) 1429
print(extent_list) 1430
print(type(extent_list)) 1431
# close the list of polygon coordinates by adding the starting node at the end again 1432
# and make list elements in the form of coordinate pairs (y,x) 1433
area_list = list([(extent[0], extent[2]),(extent[1], extent[2]),(extent[1], extent[3]),(extent[0], extent[3]),(extent[0], extent[2])]) 1434
print(area_list) 1435
print(type(area_list)) 1436
search_area = ee.Geometry.Polygon(area_list) 1437
print(search_area) 1438
print(type(search_area)) 1439
[-6.08, -6.02, 35.21, 35.31] 1440
<class 'list'> 1441
[(-6.08, 35.21), (-6.02, 35.21), (-6.02, 35.31), (-6.08, 35.31), (-6.08, 35.21)] 1442
<class 'list'> 1443
ee.Geometry({ 1444
    "functionInvocationValue": { 1445
        "functionName": "GeometryConstructors.Polygon", 1446
        "arguments": { 1447
            "coordinates": { 1448
                "constantValue": [ 1449
                    1450
                    1451
                    1452
                    1453
                    1454
                    1455
                    1456
                ] 1457
            } 1458
        } 1459
    } 1460
} 1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
26
```

```
[ 1457
[ 1458
-6.08, 1459
35.21 1460
], 1461
[ 1462
-6.02, 1463
35.21 1464
], 1465
[ 1466
-6.02, 1467
35.31 1468
], 1469
[ 1470
-6.08, 1471
35.31 1472
], 1473
[ 1474
-6.08, 1475
35.21 1476
]
] 1477
]
} 1478
},
"evenOdd": { 1480
  "constantValue": true 1481
}
} 1483
}
}) 1485
)
<class 'ee.geometry.Geometry'> 1487
16. Accessing the Sentinel-2 collection on Google Earth Engine and running the search. 1488
1489
''' 1490
----- 1491
The following block of code originates and is modified from: 1492
1493
Balzter, H. (2023) 1494
Materials for GY7709 masters computer classes. 1495
https://uniofleicester-my.sharepoint.com/personal/hb91\_leicester\_ac\_uk/\_layouts/15/onederive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-top%2FGY7709%5FSatellite%5FData%5FAnalysis% 1496
1497
1498
```

```
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper- 1499
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop% 1500
2FGY7709%5FSatellite%5FData%5FAnalysis%Fin%5FPython%2F2022%2D23&ga=1 1501
Downloaded 1 February 2023 1502
-----
''' 1503
# Obtain download links for image composites from an image collection on Google Earth 1505
Engine 1506
# Name of the Sentinel 2 image collection 1508
s2collection = ('COPERNICUS/S2') 1509
1510
# getting the median composite of Sentinel-2 images in the time range 1511
s2median = pygge.obtain_image_sentinel(s2collection, time_range, search_area, clouds) 1512
1513
# Blue, Green, Red, NIR and SWIR are downloaded 1514
bands = ['B2', 'B3', 'B4', 'B8', 'B12' ] 1515
print(bands) 1516
1517
# spatial resolution of the downloaded data 1518
resolution = 20 # in units of metres 1519
1520
# Downloading images in Geotiff, using the get_url(name, image, scale, region) method 1521
# 'region' is obtained from the area, but the format has to be adjusted using get_re- 1522
gion(geom) method 1523
search_region = pygge.get_region(search_area) 1524
s2url = pygge.get_url('s2', s2median.select(bands), resolution, search_region, filePer- 1525
Band=False) 1526
print(s2url) 1527
['B2', 'B3', 'B4', 'B8', 'B12'] 1528
https://earthengine.googleapis.com/v1alpha/projects/earthengine-legacy/thumbnails/3b62a 1529
644518dacef2cf989403d8c3e3f-44432a28d67b784f448895200db8318e:getPixels 1530
```

Downloading the data

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

```
''' 1533
----- 1534
The following block of code originates and is modified from: 1536
1537
```

Balzter, H. (2023) 1538
Materials for GY7709 masters computer classes. 1539
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15/onederive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-top%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1 1540
Downloaded 1 February 2023 1546
----- 1547
''' 1548
changing directory to download directory 1549
os.chdir(downloadaddir) 1550
1551
requesting information on the file to be downloaded 1552
f = pygge.requests.get(s2url, stream =True) 1553
1554
checking whether it is a zip file 1555
check = zipfile.is_zipfile(io.BytesIO(f.content)) 1556
1557
either download the file as is, or unzip it 1558
while not check: 1559
 f = requests.get(s2url, stream =True) 1560
 check = zipfile.is_zipfile(io.BytesIO(f.content)) 1561
else: 1562
 z = zipfile.ZipFile(io.BytesIO(f.content)) 1563
 z.extractall() 1564

Exploring the data directory structure of our downloaded files

1565
''' 1566
----- 1567
The following block of code originates and is modified from: 1568
1569
Balzter, H. (2023) 1571
Materials for GY7709 masters computer classes. 1572
https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15/onederive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-top%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1 1573
1574
1575
1576
1577

```
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1 1578
Downloaded 1 February 2023 1579
-----
''' 1580
# where we stored the downloaded Sentinel-2 images 1582
os.chdir(downloaddir) 1583
print("contents of ", downloaddir, ":") 1584
!ls -l 1585
contents of /content/work/download : 1586
total 2200 1587
-rw-r--r-- 1 root root 2251399 Apr 27 11:10 s2.tif 1588
18. the downloaded file is "s2.tif". 1589
19. the downloaded images are saved to a temporary directory that will be deleted when the virtual machine is 1590
closed. To save the images to my local directory, this is how it went; 1591
20. I went to my Google Colab folder in the panel on the left hand side. 1592
21. Found the download directory and clicked on a Sentinel-2 image folder. 1593
22. Right-clicked on it ,renamed it and selected 'download' to save it. 1594
```

Showing the after fire image of Larache as a colour composite

```
''' 1596
----- 1597
----- 1598
The following block of code originates and is modified from: 1599
1600
Balzter, H. (2023) 1601
Materials for GY7709 masters computer classes. 1602
https://uniofleicester-my.sharepoint.com/personal/hb91\_leicester\_ac\_uk/\_layouts/15/onederive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1 1603
Downloaded 1 February 2023 1609
-----
''' 1610
# getting list of all tiff files in the directory 1612
allfiles = [f for f in listdir(downloaddir) if isfile(join(downloaddir, f))] 1613
print(allfiles) 1614
```

```
1615  
# selecting the file for visualisation  
1616  
thisfile = allfiles[0]  
1617  
print(thisfile)  
1618  
['s2.tif']  
1619  
s2.tif  
1620
```

True Colour composite of after fire image of Larache

```
1621
```

```
1622  
'''  
1623  
-----  
1624
```

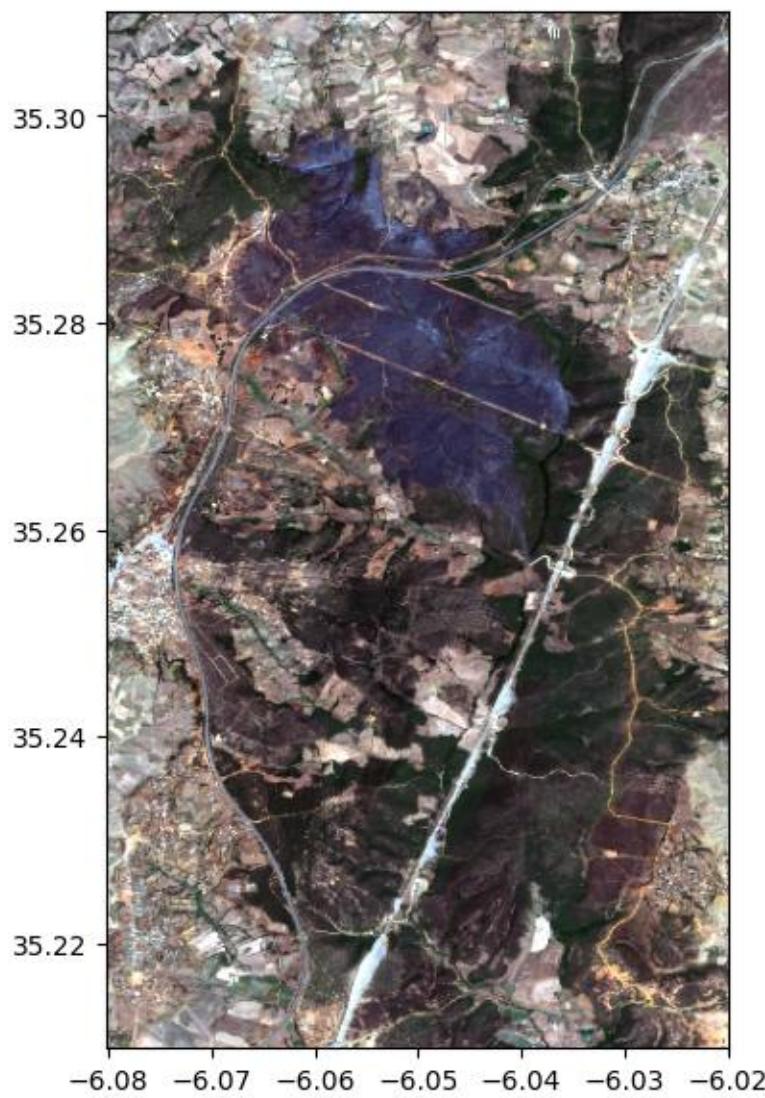
The following block of code originates and is modified from:

```
1625  
Balzter, H. (2023)  
1626  
Materials for GY7709 masters computer classes.  
1628  
https://uniofleicester-my.sharepoint.com/personal/hb91\_leicester\_ac\_uk/\_layouts/15/onederive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1  
1629  
Downloaded 1 February 2023  
1635  
-----  
1636
```

```
1637  
'''  
1638  
# creating a figure with 2x3 subplots  
1639  
fig, (ax1) = plt.subplots(1,1, figsize=(7,7))  
1640  
fig.patch.set_facecolor('white')  
1641
```

```
1642  
# the downloaded file is float32 data format  
1643  
# for plotting, uint8 data format is needed  
1644
```

```
1645  
# plotting the image with full extent  
1646  
pygge.easy_plot(thisfile, ax=ax1, bands=[3,2,1], percentiles=[0,99])  
1647  
WARNING:rasterio._env:CPLE_AppDefined in s2.tif: TIFFReadDirectory:Sum of Photometric type-related color channels and ExtraSamples doesn't match SamplesPerPixel. Defining non-color channels as ExtraSamples.  
1648  
WARNING:rasterio._env:CPLE_AppDefined in TIFFReadDirectory:Sum of Photometric type-related color channels and ExtraSamples doesn't match SamplesPerPixel. Defining non-color channels as ExtraSamples.  
1650  
1651  
1652
```



1653

False Colour Composite of after fire image of Larache

1654

'''

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/onederive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1](https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15/onederive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1)

1655

1656

1657

1658

1659

1660

1661

1662

1663

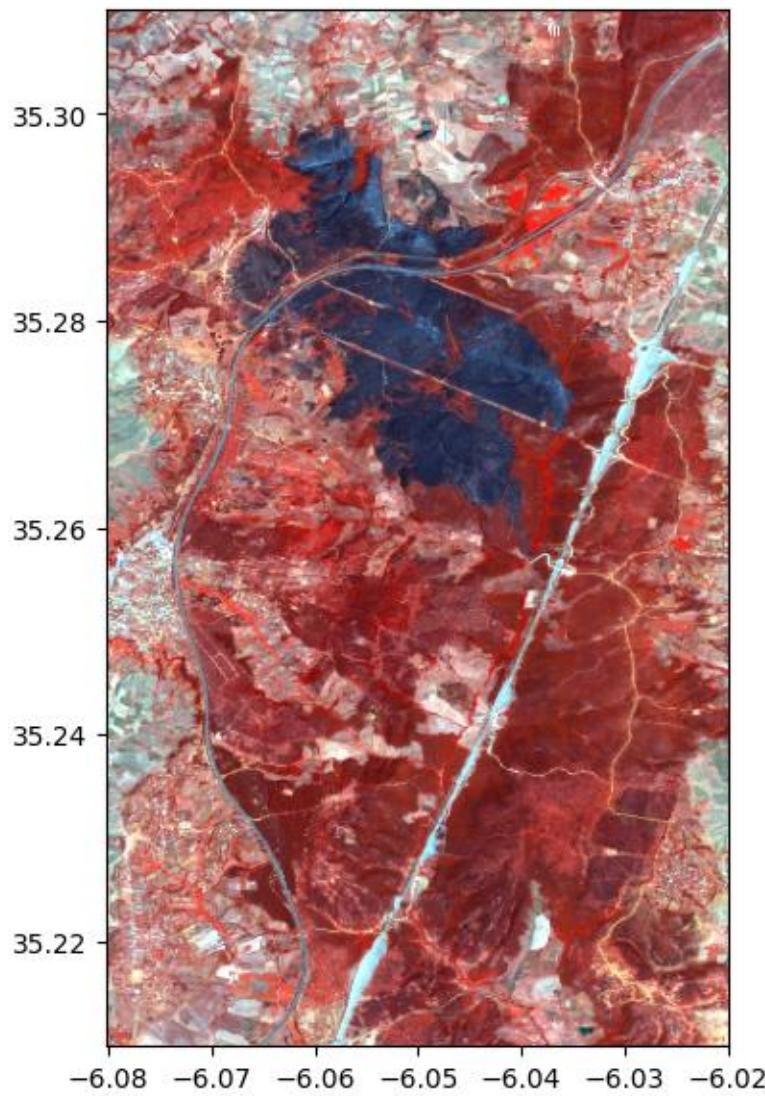
1664

1665

1666

1667

```
Downloaded 1 February 2023 1668
-----
''' 1669
# creating a figure with 2x3 subplots 1670
fig, (ax1) = plt.subplots(1,1, figsize=(7,7)) 1671
fig.patch.set_facecolor('white') 1672
# the downloaded file is float32 data format 1673
# for plotting, we need uint8 data format 1674
# plotting the image with full extent 1675
pygge.easy_plot(thisfile, ax=ax1, bands=[4,2,1], percentiles=[0,99]) 1676
WARNING:rasterio._env:CPLE_AppDefined in s2.tif: TIFFReadDirectory:Sum of Photometric type-related color channels and ExtraSamples doesn't match SamplesPerPixel. Defining non-color channels as ExtraSamples. 1677
WARNING:rasterio._env:CPLE_AppDefined in TIFFReadDirectory:Sum of Photometric type-related color channels and ExtraSamples doesn't match SamplesPerPixel. Defining non-color channels as ExtraSamples. 1678
1679
1680
1681
1682
1683
1684
1685
```



1686

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

1687

projection

1688

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

1689

1690

1691

1692

1693

The following block of code originates and is modified from:

1694

1695

Balzter, H. (2023)

1696

Materials for GY7709 masters computer classes.

1697

https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15

1698

```

/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk- 1699
top%2FGY7709%5FSatellite%5FData%5FAnalysis% 1700
5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fper- 1701
sonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop% 1702
2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1 1703
Downloaded 1 February 2023 1704
-----
''' 1705
# printing the EPSG code of our shapefile into which we want to reproject the TCI images 1706
print("Reprojecting image to EPSG projection ", epsg) 1707
1708
# making a file name for our new file 1709
warpfile = thisfile.split(sep='.')[0] + '_warped.tif' 1710
print("We are in this directory: ") 1711
!pwd 1712
print("Input file: ", thisfile) 1713
print("Output file: ", warpfile) 1714
1715
# calling the easy_warp function 1716
tmp = pygge.easy_warp(thisfile, warpfile, epsg) 1717
Reprojecting image to EPSG projection 4326 1718
We are in this directory: 1719
/content/work/download 1720
WARNING:rasterio._env:CPLE_AppDefined in s2.tif: TIFFReadDirectory:Sum of Photometric type-related color channels and ExtraSamples doesn't match SamplesPerPixel. Defining non-color channels as ExtraSamples. 1721
WARNING:rasterio._env:CPLE_AppDefined in TIFFReadDirectory:Sum of Photometric type-related color channels and ExtraSamples doesn't match SamplesPerPixel. Defining non-color channels as ExtraSamples. 1722
Input file: s2.tif 1723
Output file: s2_warped.tif 1724
Creating warped file:s2_warped.tif 1725

```

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.

```

''' 1733
----- 1734
The following block of code originates and is modified from: 1735
----- 1736

```

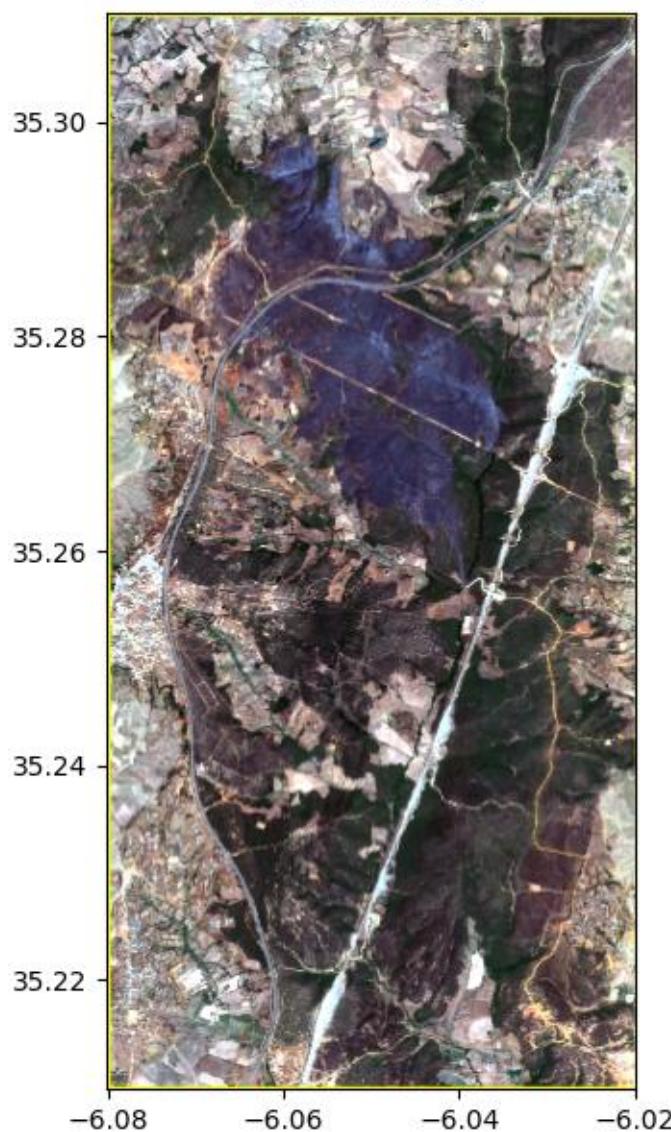
Balzter, H. (2023) Materials for GY7709 masters computer classes. [https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15/onederive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2F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

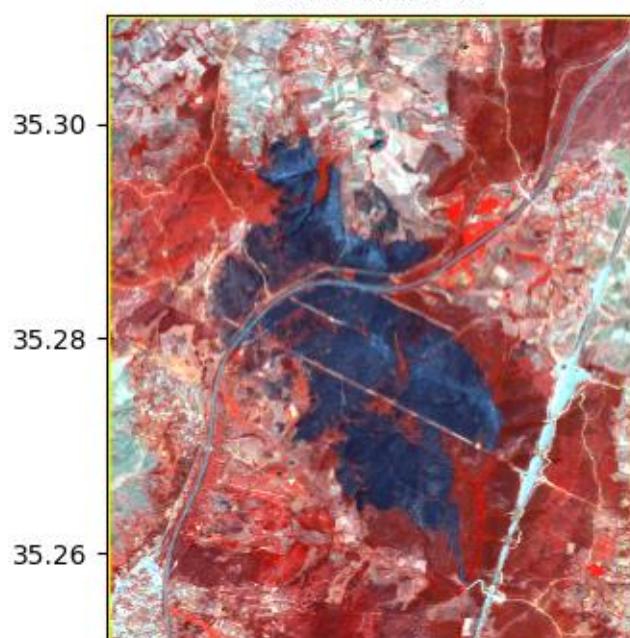
plotting the image with full extent
pygge.easy_plot(warpfile, ax=ax[0], percentiles=[0,98], bands=[3,2,1],
shapefile=shapefile, fillcolor="none", linecolor="yellow",
title="LARACHE AFTER FIRE")

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")
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766

LARACHE AFTER FIRE



LARACHE AFTER FIRE



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

1768

ITS METADATA

1769

```
'''  
-----  
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%2Ezip&parent=%2Fpersonal%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 coordinate reference system  
larache_before = rasterio.open(join(rootdir + "/larache_before/before_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["EPSG","9122"]],AXIS["Latitude",NORTH],AXIS["Longitude",EAST],AUTHORITY["EPSG","4326"]]' )
```

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

1799

METADATA

1800

```
'''  
-----  
1801  
1802  
1803
```

The following block of code originates and is modified from:

```

1804
1805
Balzter, H. (2023) 1806
Materials for GY7709 masters computer classes. 1807
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%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1 1808
1809
1810
1811
1812
1813
1814
Downloaded 1 February 2023 1815
-----
1815
1816
''' 1816
# reading in the warped after fire downloaded image of larache and checking its coordinate system 1817
1818
larache_after = rasterio.open(join(rootdir + "/larache_after/after_fire_warped.tif")) 1819
src2=larache_after 1820
d2=src2.crs.to_wkt() 1821
print(type(d2)) 1822
pprint(d2) 1823
<class 'str'> 1824
('GEOGCS["WGS 84",DATUM["WGS_1984",SPHEROID["WGS ' 1825
'84",6378137,298.257223563,AUTHORITY["EPSG","7030"]],AUTHORITY["EPSG","6326"]],PRIMEM[ 1826
"Greenwich",0,AUTHORITY["EPSG","8901"]],UNIT["degree",0.0174532925199433,AUTHORITY["EPS 1827
G","9122"]],AXIS["Latitude",NORTH],AXIS["Longitude",EAST],AUTHORITY["EPSG","4326"]]' 1828

```

CHECKING IF THE BEFORE AND AFTER DATASETS ARE OF SAME SHAPE

```

1829
1830
# checking if the before and after image data are the same shape?
1831
larache_before.shape == larache_after.shape 1832
1833

```

True

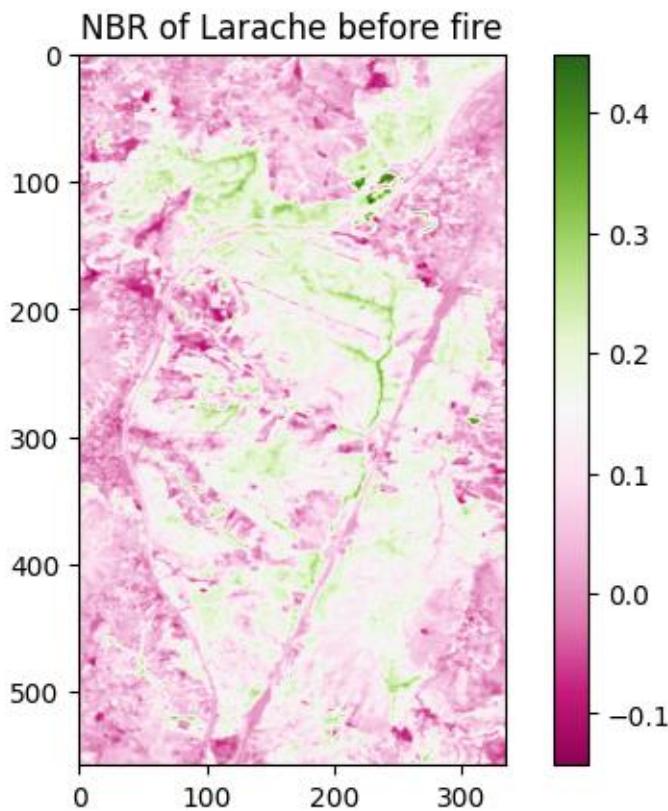
CALCULATION OF NBR FOR THE BEFORE FIRE IMAGE OF LARACHE

```

1835
1836
''' 1837
-----
1838
The following block of code originates and is modified from: 1839
1840

```

Balzter, H. (2023) 1841
Materials for GY7709 masters computer classes. 1842
[https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15/onederive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-top%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1](https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15/onederive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesk-top%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1) 1843
Downloaded 1 February 2023 1844
----- 1845
''' 1851
swir_before = larache_before.read(5) # band in position 5 in our stacked image i.e SWIR 1852
nir_before = larache_before.read(4) # band in position 4 in our stacked image i.e NIR 1853
1854
calculation of NBR before the fire as floating point array 1855
nbr_before = (nir_before.astype(float) - swir_before.astype(float)) / (nir_before.astype(float) + swir_before.astype(float)) 1856
1857
Ignoring division by zero 1858
np.seterr(divide='ignore', invalid='ignore') 1859
1860
printing some image statistics, ignoring missing values (nan) 1861
print("minimum NBR_before = ", np.nanmin(nbr_before)) 1862
print("mean NBR_before = ", np.nanmean(nbr_before)) 1863
print("maximum NBR_before = ", np.nanmax(nbr_before)) 1864
print("standard deviation = ", np.nanstd(nbr_before)) 1865
minimum NBR_before = -0.14298390392538413 1866
mean NBR_before = 0.10239791290152904 1867
maximum NBR_before = 0.44895833342572644 1868
standard deviation = 0.0735262823032699 1869
1870
PLOT OF NBR BEFORE THE FIRE 1871
1872
plt.imshow(nbr_before, cmap='PiYG') # colours to use include Purple and Green 1873
plt.colorbar() 1874
plt.title('NBR of Larache before fire') # title of plot 1875
plt.show() # plot 1876
1877



1878

CALCULATION OF NBR FOR THE AFTER FIRE IMAGE OF LARACHE

1879

'''

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/onederive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1](https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15/onederive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fpersonal%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

1896

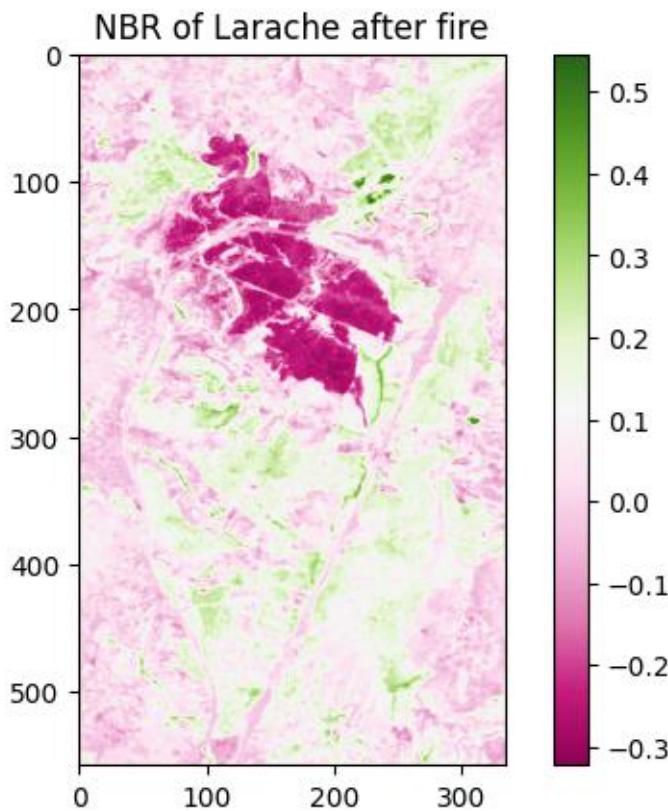
1897

1898

```
# calculation of NBR after the fire as floating point array 1899
nbr_after = (nir_after.astype(float) - swir_after.astype(float)) / (nir_af- 1900
ter.astype(float) + swir_after.astype(float)) 1901
1902
# Ignoring division by zero 1903
np.seterr(divide='ignore', invalid='ignore') 1904
1905
# printing some image statistics, ignoring missing values (nan) 1906
print("minimum NBR_after = ", np.nanmin(nbr_after)) 1907
print("mean NBR_after = ", np.nanmean(nbr_after)) 1908
print("maximum NBR_after = ", np.nanmax(nbr_after)) 1909
print("standard deviation = ", np.nanstd(nbr_after)) 1910
minimum NBR_after = -0.3224225354386313 1911
mean NBR_after = 0.049028595859026615 1912
maximum NBR_after = 0.5460599474035327 1913
standard deviation = 0.11624440093712922 1914
```

PLOT OF NBR AFTER THE FIRE

```
1915
plt.imshow(nbr_after, cmap='PiYG') # Purple and Green selected as colours in plot 1916
plt.colorbar() 1917
plt.title('NBR of Larache after fire') # title of plot 1918
plt.show() # display plot or map 1919
1920
```



1921

CALCULATION OF dNBR AS DIFFERENCE OF BEFORE AND AFTER NBR RASTERS

1922

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%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1](https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1)

Downloaded 1 February 2023

```
'''
```

1923

1924

1925

1926

1927

1928

1929

1930

1931

1932

1933

1934

1935

1936

1937

1938

1939

1940

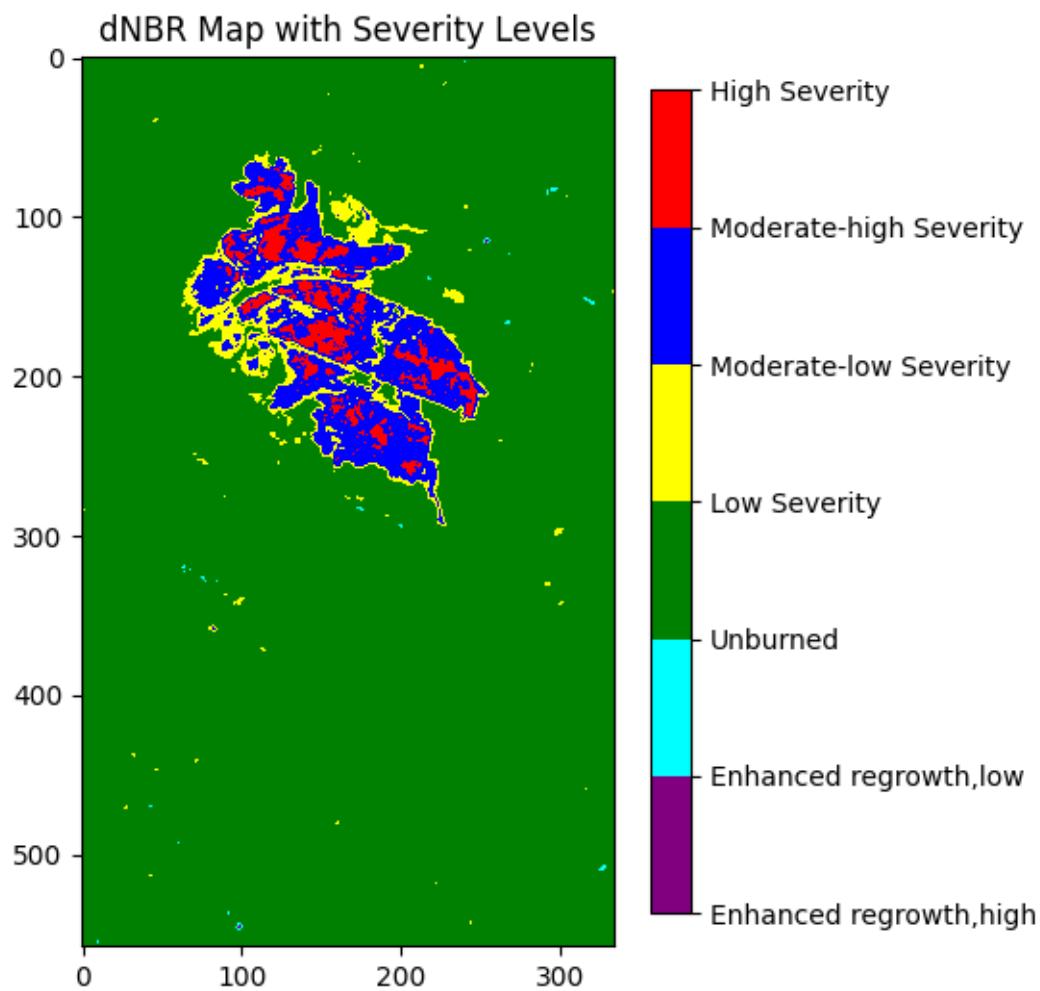
1941

```
1942  
# printing some image statistics, ignoring missing values (nan)  
1943 print("minimum dNBR = ", np.nanmin(dNBR))  
1944 print("mean dNBR = ", np.nanmean(dNBR))  
1945 print("maximum dNBR = ", np.nanmax(dNBR))  
1946 print("standard deviation = ", np.nanstd(dNBR))  
1947 minimum dNBR = -0.363571844669598  
1948 mean dNBR = 0.05336931704250243  
1949 maximum dNBR = 0.5638484811508442  
1950 standard deviation = 0.10752007563079775  
1951
```

MAP SHOWING AREAS WHERE THE FIRE DAMAGE HAS BEEN GREATEST

```
1952  
  
1953  
# Defining the dNBR values and the corresponding severity levels  
1954 dNBR  
1955  
1956  
1957 levels = [-0.5, -0.25, -0.1, 0.1, 0.27, 0.44, 0.66]  
1958 labels = ['Enhanced regrowth,high', 'Enhanced regrowth,low', 'Unburned', 'Low Severity',  
1959 'Moderate-low Severity', 'Moderate-high Severity', 'High Severity']  
1960  
1961 # Creating a custom color map with 5 colors  
1962 cmap = colors.ListedColormap(['purple', 'cyan', 'green', 'yellow', 'blue', 'darkred',  
1963 'red'])  
1964  
1965 # Setting the boundaries and norm of the color map  
1966 bounds = [-0.5, -0.25, -0.1, 0.1, 0.27, 0.44, 0.66]  
1967 norm = colors.BoundaryNorm(bounds, cmap.N)  
1968  
1969 # Creating a figure and axis object  
1970 fig, ax = plt.subplots(figsize=(6, 6))  
1971  
1972 # Plotting the dNBR using the custom color map and norm  
1973 im = ax.imshow(dNBR, cmap=cmap, norm=norm)  
1974  
1975 # Adding a color bar with custom tick labels  
1976 cbar = fig.colorbar(im, ax=ax, fraction=0.046, pad=0.04, boundaries=bounds, ticks=levels)  
1977 cbar.ax.set_yticklabels(labels)  
1978  
1979 # Setting the title and axis labels  
1980 ax.set_title('dNBR Map with Severity Levels')  
1981
```

```
plt.show()
```



CALCULATION OF AREAS OF BURNED CLASSES USING dNBR

1985

```
# 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))
```

1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998

```

# Calculate the area of each class or label in hectares 1999
regrowth_high_area = regrowth_high_pixels * pixel_area 2000
regrowth_low_area = regrowth_low_pixels * pixel_area 2001
unburned_area = unburned_pixels * pixel_area 2002
low_severity_area = low_severity_pixels * pixel_area 2003
mod_low_severity_area = mod_low_severity_pixels * pixel_area 2004
mod_high_severity_area = mod_high_severity_pixels * pixel_area 2005
high_severity_area = high_severity_pixels * pixel_area 2006

# Print the area of each class or label 2007
print('Enhanced regrowth, high area:', regrowth_high_area, 'hectares') 2008
print('Enhanced regrowth, low area:', regrowth_low_area, 'hectares') 2009
print('Unburned area:', unburned_area, 'hectares') 2010
print('Low severity area:', low_severity_area, 'hectares') 2011
print('Moderate-low severity area:', mod_low_severity_area, 'hectares') 2012
print('Moderate-high severity area:', mod_high_severity_area, 'hectares') 2013
print('High severity area:', high_severity_area, 'hectares') 2014
Enhanced regrowth, high area: 0.003600000000000003 hectares 2015
Enhanced regrowth, low area: 0.0664 hectares 2016
Unburned area: 74.6992 hectares 2017
Low severity area: 2.0148 hectares 2018
Moderate-low severity area: 4.2816 hectares 2019
Moderate-high severity area: 1.4420000000000002 hectares 2020
High severity area: 0.0 hectares 2021

```

CALCULATION OF AVERAGE NBR2 FOR BURNED POLYGON

```

# Opening the TIF file and the shapefile of burned polygon 2024
with rasterio.open('/content/drive/MyDrive/satelliteCW1/larache_after/af- 2025
ter_fire_warped.tif') as src: 2026
    shapes = gpd.read_file('/content/drive/MyDrive/satelliteCW1/burned_layers/POLY- 2027
GON.shp') 2028

# Masking the TIF file using the shapefile 2029
masked_data, _ = rasterio.mask.mask(src, shapes.geometry, crop=True) 2030

# Calculating the NBR2_burned 2031
# the NIR band has index 3 and SWIR band has index 4 2032
NBR2_burned = (masked_data[3] - masked_data[4]) / (masked_data[3] + masked_data[4]) 2033

# Ignoring division by zero 2034

```

```

np.seterr(divide='ignore', invalid='ignore') 2039
2040
# print average NBR2 for burned polygon, ignoring missing values (nan) 2041
2042
print("average NBR2_burned = ", np.nanmean(NBR2_burned)) 2043
average NBR2_burned = -0.13870606 2044

```

CALCULATION OF AVERAGE NBR1 FOR BURNED POLYGON

```

2045
2046
# Opening the TIF file and the shapefile of burned polygon 2047
with rasterio.open('/content/drive/MyDrive/satelliteCW1/larache_before/be- 2048
fore_fire_warped.tif') as src: 2049
    shapes = gpd.read_file('/content/drive/MyDrive/satelliteCW1/burned_layers/POLY- 2050
GON.shp') 2051
2052
# Masking the TIF file using the shapefile 2053
masked_data, _ = rasterio.mask.mask(src, shapes.geometry, crop=True) 2054
2055
# Calculating the NBR1_burned 2056
# the NIR band has index 3 and SWIR band has index 4 2057
NBR1_burned = (masked_data[3] - masked_data[4]) / (masked_data[3] + masked_data[4]) 2058
2059
# Ignoring division by zero 2060
np.seterr(divide='ignore', invalid='ignore') 2061
2062
# printing average NBR1 for burned polygon, ignoring missing values (nan) 2063
2064
print("average NBR1_burned = ", np.nanmean(NBR1_burned)) 2065
average NBR1_burned = 0.1715795 2066

```

25. CALCULATION OF dNBR FOR BURNED POLYGON

```

2067
2068
# finding the difference between the NBR before and after the fire 2069
dNBR_burned = NBR1_burned - NBR2_burned 2070
2071
# printing average dNBR for burned polygon, ignoring missing values (nan) 2072
2073
print("average dNBR_burned = ", np.nanmean(dNBR_burned)) 2074
average dNBR_burned = 0.3102856 2075

```

CALCULATION OF AVERAGE NBR1 FOR UNBURNED POLYGON

2076

```

2077
# Opening the TIF file and the shapefile of unburned polygon
2078
with rasterio.open('/content/drive/MyDrive/satelliteCW1/larache_before/before_fire_warped.tif') as src:
2079
    shapes = gpd.read_file('/content/drive/MyDrive/satelliteCW1/unburned_layers/POLY-
2080 GON.shp')
2081
2082
    # Masking the TIF file using the shapefile
2083
    masked_data, _ = rasterio.mask.mask(src, shapes.geometry, crop=True)
2084
2085
    # Calculating the NBR1_unburned
2086
    # the NIR band has index 3 and SWIR band has index 4
2087
    NBR1_unburned = (masked_data[3] - masked_data[4]) / (masked_data[3] +
2088 masked_data[4])
2089
2090
    # Ignoring division by zero
2091
    np.seterr(divide='ignore', invalid='ignore')
2092
2093
# printing average NBR1 for unburned poloygon, ignoring missing values (nan)
2094
2095
print("average NBR1_unburned = ", np.nanmean(NBR1_unburned))
2096
average NBR1_unburned = 0.17309634
2097
2098

```

CALCULATION OF AVERAGE NBR2 FOR UNBURNED POLYGON

```

2099
2100
# Opening the TIF file and the shapefile of unburned polygon
2101
with rasterio.open('/content/drive/MyDrive/satelliteCW1/larache_after/af-
2102 ter_fire_warped.tif') as src:
2103
    shapes = gpd.read_file('/content/drive/MyDrive/satelliteCW1/unburned_layers/POLY-
2104 GON.shp')
2105
2106
    # Masking the TIF file using the shapefile
2107
    masked_data, _ = rasterio.mask.mask(src, shapes.geometry, crop=True)
2108
2109
    # Calculating the NBR2_unburned
2110
    # the NIR band has index 3 and SWIR band has index 4
2111
    NBR2_unburned = (masked_data[3] - masked_data[4]) / (masked_data[3] +
2112 masked_data[4])
2113
2114
    # Ignoring division by zero
2115
    np.seterr(divide='ignore', invalid='ignore')
2116

```

```
2117  
# printing average NBR2 for unburned poloygon, ignoring missing values (nan) 2118  
2119  
print("average NBR2_unburned = ", np.nanmean(NBR2_unburned))  
average NBR2_unburned = 0.17163754 2120  
2121
```

CALCULATION OF dNBR FOR UNBURNED POLYGON

```
2122  
2123  
# finding the difference between the NBR before and after the fire 2124  
dNBR_unburned = NBR1_unburned - NBR2_unburned 2125  
2126  
# printing average dNBR for burned polygon, ignoring missing values (nan) 2127  
2128  
print("average dNBR_unburned = ", np.nanmean(dNBR_unburned))  
average dNBR_unburned = 0.0014588113 2129  
2130
```

CALCULATION OF NDVI FOR THE BEFORE FIRE IMAGE OF LARACHE

```
2131  
2132  
'''  
-----  
The following block of code originates and is modified from:  
2133  
2134  
Balzter, H. (2023)  
Materials for GY7709 masters computer classes.  
https://uniofleicester-my.sharepoint.com/personal/hb91\_leicester\_ac\_uk/\_layouts/15/onederive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1  
Downloaded 1 February 2023  
-----
```

```
2145  
2146  
'''  
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  
2147  
# calculation of NDVI before the fire as flaoting point array  
NDVI_before = (nir_before.astype(float) - r_before.astype(float)) / (nir_before.astype(float) + r_before.astype(float))  
2150  
2151  
2152  
2153
```

```
2154  
# Ignore division by zero  
2155 np.seterr(divide='ignore', invalid='ignore')  
2156  
2157 # print some image statistics, ignoring missing values (nan)  
2158 print("minimum NDVI_before = ", np.nanmin(NDVI_before))  
2159 print("mean NDVI_before = ", np.nanmean(NDVI_before))  
2160 print("maximum NDVI_before = ", np.nanmax(NDVI_before))  
2161 print("standard deviation = ", np.nanstd(NDVI_before))  
2162 minimum NDVI_before = -0.0031236251459653844  
2163 mean NDVI_before = 0.16960801456931313  
2164 maximum NDVI_before = 0.44798869776436046  
2165 standard deviation = 0.053921411038247144  
2166
```

CALCULATION OF NDVI FOR THE AFTER FIRE IMAGE OF LARACHE

```
2167  
'''  
-----  
The following block of code originates and is modified from:  
2168
```

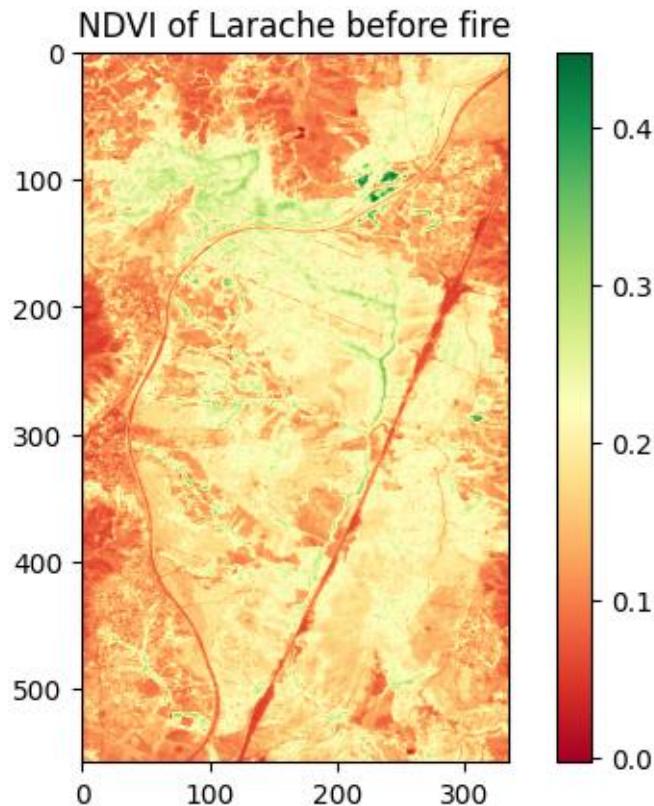
```
2169  
Balzter, H. (2023)  
2170 Materials for GY7709 masters computer classes.  
2171  
https://uniofleicester-my.sharepoint.com/personal/hb91\_leicester\_ac\_uk/\_layouts/15/onederive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1  
2172  
Downloaded 1 February 2023  
-----  
'''
```

```
2173  
r_after = larache_after.read(3) # band 3 in our stacked image is Red  
2174 nir_after = larache_after.read(4) # band 4 in our stacked image is NIR  
2175  
2176 # calculation of NDVI after the fire as floating point array  
2177 NDVI_after = (nir_after.astype(float) - r_after.astype(float)) / (nir_after.astype(float) + r_after.astype(float))  
2178  
2179 # Ignoring division by zero  
2180 np.seterr(divide='ignore', invalid='ignore')  
2181  
2182  
2183  
2184  
2185  
2186  
2187  
2188  
2189  
2190  
2191  
2192  
2193
```

```
# print some image statistics, ignoring missing values (nan) 2194
print("minimum NDVI_after = ", np.nanmin(NDVI_after)) 2195
print("mean NDVI_after = ", np.nanmean(NDVI_after)) 2196
print("maximum NDVI_after = ", np.nanmax(NDVI_after)) 2197
print("standard deviation = ", np.nanstd(NDVI_after)) 2198
minimum NDVI_after = -0.06513785137467486 2199
mean NDVI_after = 0.17277971166410844 2200
maximum NDVI_after = 0.5750123655099424 2201
standard deviation = 0.07998356629775627 2202
```

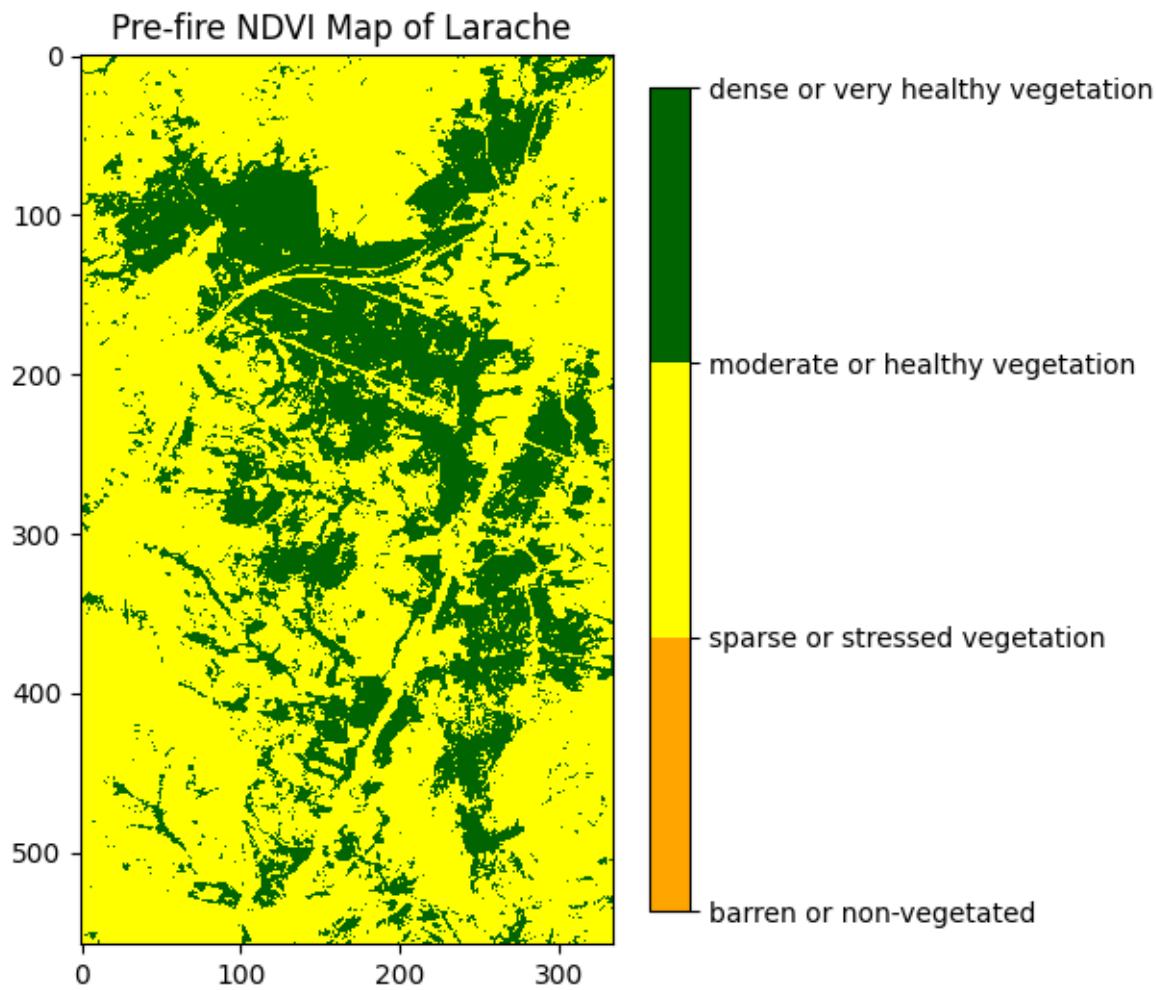
MAPS OF NDVI BEFORE THE FIRE

```
2203
plt.imshow(NDVI_before, cmap='RdYlGn') # selection of Red to Green colours 2204
plt.colorbar() # colour bar 2205
plt.title('NDVI of Larache before fire') # map title 2206
plt.show() # showing plot 2207
2208
```



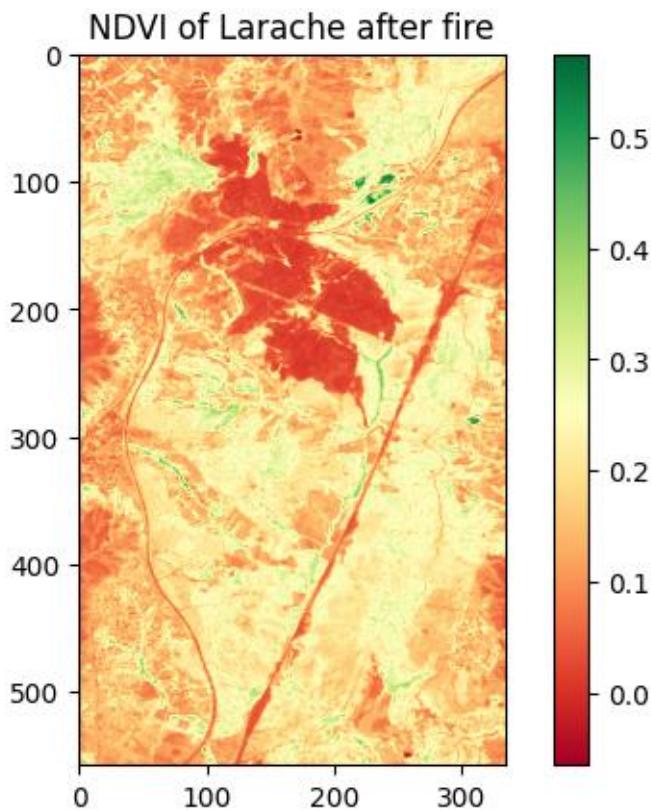
```
# Defining the pre-fire NDVI levels 2209
NDVI_before 2210
levels = [-1, 0, 0.2, 0.5] 2211
2212
2213
2214
```

```
labels = ['barren or non-vegetated', 'sparse or stressed vegetation', 'moderate or  
healthy vegetation', 'dense or very healthy vegetation'] 2215  
2216  
2217  
# Creating a custom color map with 5 colors 2218  
cmap = colors.ListedColormap(['orange', 'yellow', 'lightgreen', 'darkgreen']) 2219  
2220  
# Setting the boundaries and norm of the color map 2221  
bounds = [-1, 0, 0.2, 0.5] 2222  
norm = colors.BoundaryNorm(bounds, cmap.N) 2223  
2224  
# Creating a figure and axis object 2225  
fig, ax = plt.subplots(figsize=(6, 6)) 2226  
2227  
# Plotting the pre-fire NDVI using the custom color map and norm 2228  
im = ax.imshow(NDVI_before, cmap=cmap, norm=norm) 2229  
2230  
# Adding a color bar with custom tick labels 2231  
cbar = fig.colorbar(im, ax=ax, fraction=0.046, pad=0.04, boundaries=bounds, ticks=levels)  
cbar.ax.set_yticklabels(labels) 2234  
2235  
# Setting the title and axis labels 2236  
ax.set_title('Pre-fire NDVI Map of Larache ') 2237  
2238  
plt.show() 2239
```



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 2249
NDVI_after 2250

levels = [-1, 0, 0.2, 0.5] 2251
labels = ['barren or non-vegetated', 'sparse or stressed vegetation', 'moderate or 2252
healthy vegetation', 'dense or very healthy vegetation'] 2253

# Creating a custom color map with 5 colors 2254
cmap = colors.ListedColormap(['orange', 'yellow', 'lightgreen', 'darkgreen']) 2255

# Setting the boundaries and norm of the color map 2256
bounds = [-1, 0, 0.2, 0.5] 2257
norm = colors.BoundaryNorm(bounds, cmap.N) 2258

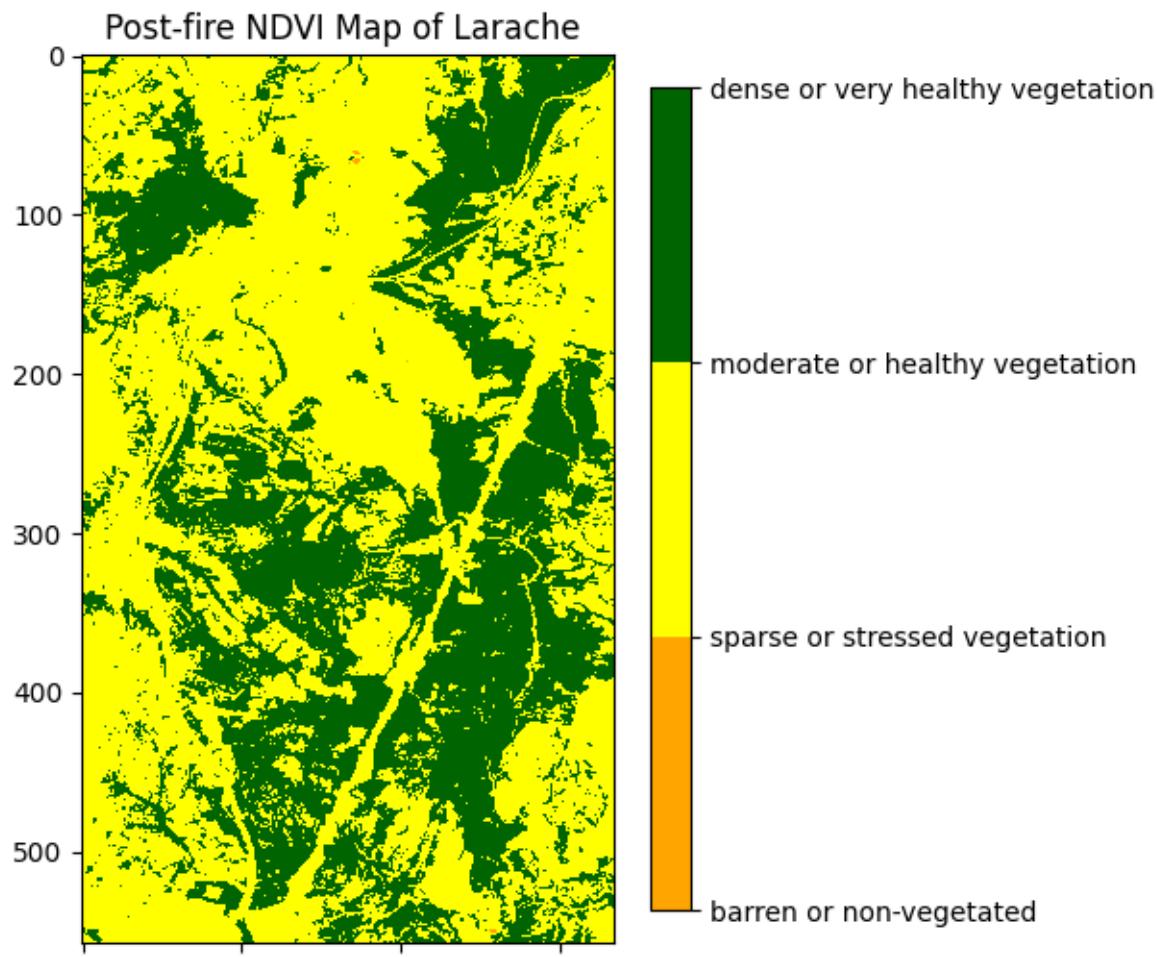
# Creating a figure and axis object 2259
fig, ax = plt.subplots(figsize=(6, 6)) 2260

# Plotting the post-fire NDVI using the custom color map and norm 2261
im = ax.imshow(NDVI_after, cmap=cmap, norm=norm) 2262

# Adding a color bar with custom tick labels 2263
cbar = fig.colorbar(im, ticks=[-1, 0, 0.2, 0.5], 2264
                    label='Normalized Difference Vegetation Index') 2265

# Saving the figure 2266
fig.savefig('post_fire_ndvi.png') 2267
```

```
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%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1](https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fpersonal%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

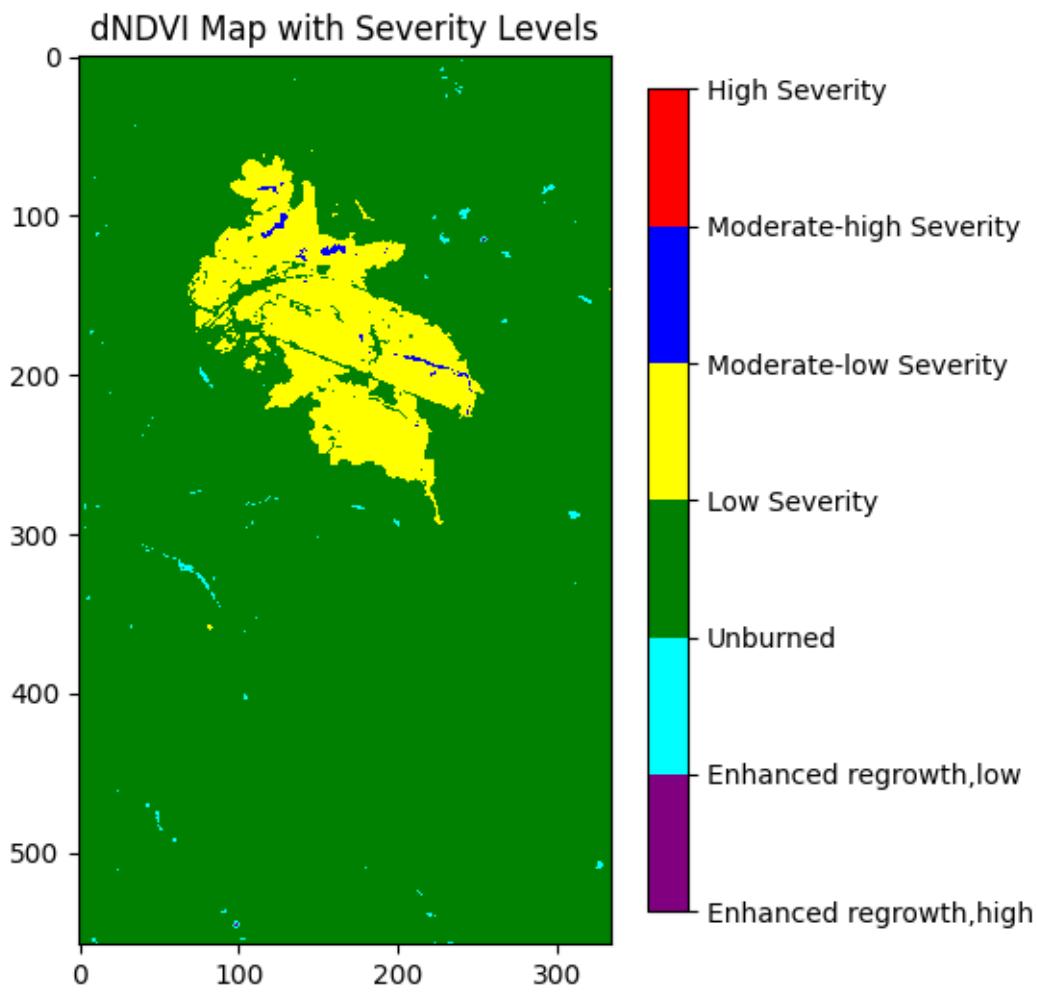
levels = [-0.5, -0.25, -0.1, 0.1, 0.27, 0.44, 0.66]
labels = ['Enhanced regrowth,high','Enhanced regrowth,low','Unburned', 'Low Severity',
'Moderate-low Severity', 'Moderate-high Severity', 'High Severity']

# 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)) 2327  
2328  
# Plotting the dNDVI using the custom color map and norm  
im = ax.imshow(dNDVI, cmap=cmap, norm=norm) 2329  
2330  
# Adding a color bar with custom tick labels  
cbar = fig.colorbar(im, ax=ax, fraction=0.046, pad=0.04, boundaries=bounds, ticks=levels) 2331  
2332  
cbar.ax.set_yticklabels(labels) 2333  
2334  
# Setting the title and axis labels  
ax.set_title('dNDVI Map with Severity Levels') 2335  
2336  
plt.show() 2337  
2338  
2339  
2340
```



CALCULATION OF BURNED AREAS BASED ON dNDVI

2342

2343

```

# Define the area of each pixel in hectares assuming a spatial resolution of 20 meters 2344
pixel_area = 0.0004 2345
2346

# Calculate the number of pixels in each class or label 2347
regrowth_high_pixels = np.sum((dNDVI >= -0.5) & (dNDVI < -0.251)) 2348
regrowth_low_pixels = np.sum((dNDVI >= -0.25) & (dNDVI < -0.101)) 2349
unburned_pixels = np.sum((dNDVI >= -0.100) & (dNDVI < 0.99)) 2350
low_severity_pixels = np.sum((dNDVI >= 0.100) & (dNDVI < 0.269)) 2351
mod_low_severity_pixels = np.sum((dNDVI >= 0.270) & (dNDVI < 0.439)) 2352
mod_high_severity_pixels = np.sum((dNDVI >= 0.44) & (dNDVI < 0.659)) 2353
high_severity_pixels = np.sum((dNDVI >= 0.660) & (dNDVI < 1.300)) 2354
2355

# Calculate the area of each class or label in hectares 2356
regrowth_high_area = regrowth_high_pixels * pixel_area 2357
regrowth_low_area = regrowth_low_pixels * pixel_area 2358
unburned_area = unburned_pixels * pixel_area 2359
low_severity_area = low_severity_pixels * pixel_area 2360
mod_low_severity_area = mod_low_severity_pixels * pixel_area 2361
mod_high_severity_area = mod_high_severity_pixels * pixel_area 2362
high_severity_area = high_severity_pixels * pixel_area 2363
2364

# Print the area of each class or label 2365
print('Enhanced regrowth, high area:', regrowth_high_area, 'hectares') 2366
print('Enhanced regrowth, low area:', regrowth_low_area, 'hectares') 2367
print('Unburned area:', unburned_area, 'hectares') 2368
print('Low severity area:', low_severity_area, 'hectares') 2369
print('Moderate-low severity area:', mod_low_severity_area, 'hectares') 2370
print('Moderate-high severity area:', mod_high_severity_area, 'hectares') 2371
print('High severity area:', high_severity_area, 'hectares') 2372
Enhanced regrowth, high area: 0.004 hectares 2373
Enhanced regrowth, low area: 0.2932 hectares 2374
Unburned area: 74.45960000000001 hectares 2375
Low severity area: 6.163600000000001 hectares 2376
Moderate-low severity area: 0.15760000000000002 hectares 2377
Moderate-high severity area: 0.0 hectares 2378
High severity area: 0.0 hectares 2379

```

TESTING WHETHER THE dNBR WAS CORRELATED TO dNDVI

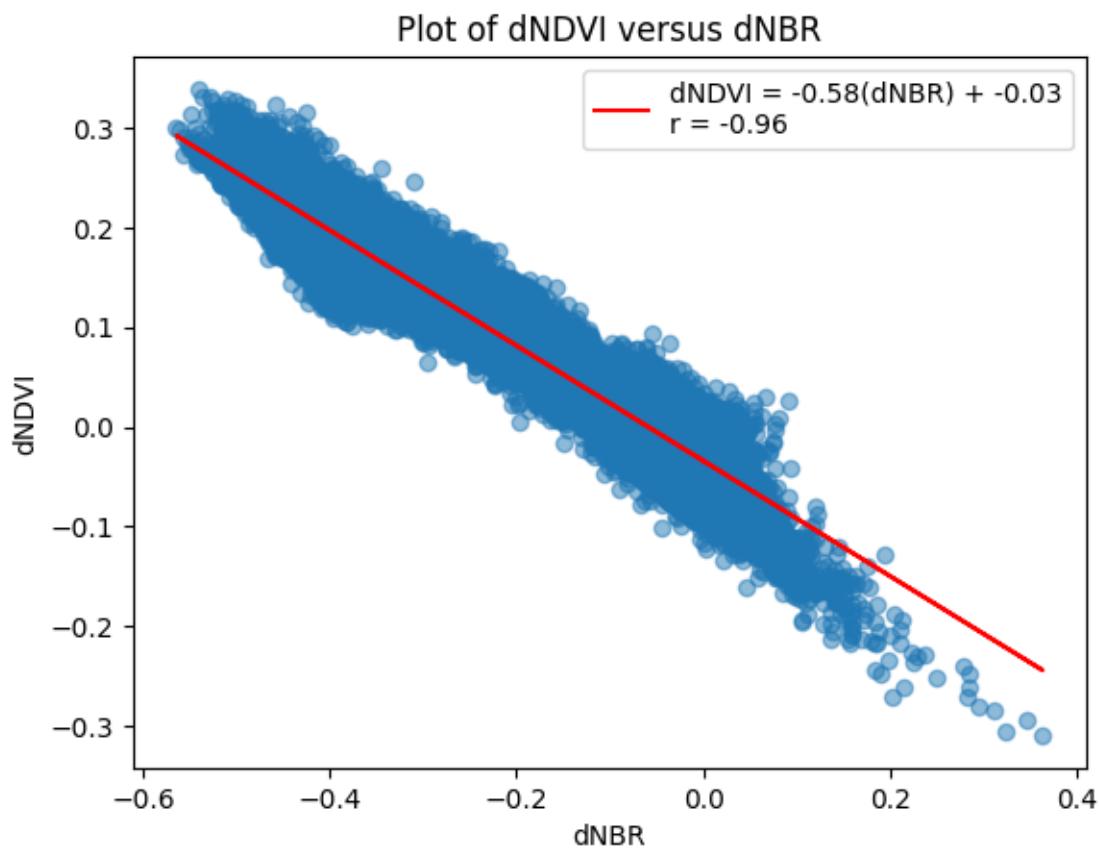
2380

2381

2382

2383

```
2384  
# dNBR and dNDVI are 2D arrays with the same shape  
2385  
dNBR_flat = dNBR.flatten()  
2386  
dNDVI_flat = dNDVI.flatten()  
2387  
2388  
# Calculate the linear regression line  
2389  
slope, intercept, r_value, p_value, std_err = linregress(dNBR_flat, dNDVI_flat)  
2390  
line = slope * dNBR_flat + intercept  
2391  
2392  
# Create a scatter plot  
2393  
fig, ax = plt.subplots()  
2394  
ax.scatter(dNBR_flat, dNDVI_flat, alpha=0.5)  
2395  
2396  
# Add the linear regression line to the scatter plot with the equation and correlation  
2397 coefficient  
2398  
eqn_label = f'dNDVI = {slope:.2f}(dNBR) + {intercept:.2f}'  
2399  
corr_label = f'r = {r_value:.2f}'  
2400  
ax.plot(dNBR_flat, line, color='red', label=f'{eqn_label}\n{corr_label}')  
2401  
2402  
# Set the x-axis and y-axis labels  
2403  
ax.set_xlabel('dNBR')  
2404  
ax.set_ylabel('dNDVI')  
2405  
2406  
# Set the title of the plot  
2407  
ax.set_title('Plot of dNDVI versus dNBR')  
2408  
2409  
# Add the legend to the plot  
2410  
ax.legend()  
2411  
2412  
plt.show()  
2413
```



2414

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%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2Ezip&parent=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23&ga=1](https://uniofleicester-my.sharepoint.com/personal/hb91_leicester_ac_uk/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fhb91%5Fleicester%5Fac%5Fuk%2FDocuments%2FDesktop%2FGY7709%5FSatellite%5FData%5FAnalysis%5Fin%5FPython%2F2022%2D23%2FGY7709%2D2022%2D23%2Ezip&parent=%2Fpersonal%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:

2415

2416

2417

2418

2419

2420

2421

2422

2423

2424

2425

2426

2427

2428

2429

2430

2431

2432

2433

```
!pip install -U notebook-as-pdf          2434
!pypeteer-install                         2435
!jupyter nbconvert /content/drive/MyDrive/satelliteCW1/229010645_GY7709_CW2.ipynb --to
html                                     2436
                                         2437
                                         2438
```