

EÖTVÖS LORÁND UNIVERSITY  
FACULTY OF INFORMATICS  
DEPARTMENT OF CARTOGRAPHY AND GEOINFORMATICS

# **GeeFire: A Web Application for Forest Fire Severity Assessment using Google Earth Engine**

**Mary Ruth Bongon**

student of Cartography MSc

**Dr. Mátyás Gede**

Associate professor

ELTE Institute of Cartography and Geoinformatics



Budapest, 2023

## **Acknowledgement**

Sometimes, living feels like a wood drifting into the vastness of the ocean. More often, the sun is shining and the days are calm. However, there are days when the storm is trying to drown you, so you're silently hoping someone is there to save you.

To the people who are there for me through thick and thin, saving me from my thoughts that are drowning me, this is all for you.

To my amazing, super cool, and very supportive supervisor, Dr. Gede, thank you so much for making my thesis days bearable. Your willingness to listen to my ideas and your enthusiasm for teaching me and pointing me in the right direction made a huge impact on finishing this thesis.

To the sunshine in my rainy days, Zsolty, my love, no words can ever truly express how grateful I am to you. The last few weeks before the thesis deadline were tough, and I couldn't thank you enough for all the little things you do for me. Spending the mundanity of life with you gives me a sense of purpose.

And lastly, to my family, especially my grandma, this is for you. I hope I make you proud again!

MRB

# Table Of Contents

<b>Acknowledgement</b>	<b>2</b>
<b>Table Of Contents</b>	<b>3</b>
<b>List of Tables</b>	<b>5</b>
<b>List of Figures</b>	<b>6</b>
<b>Abstract</b>	<b>7</b>
<b>Chapter I</b>	<b>8</b>
<b>Introduction</b>	<b>8</b>
1.1 Introduction and General Background	8
1.2 Research Problem	10
1.3 Scope and Delimitation	11
1.4 Significance of the Study	12
<b>Chapter II</b>	<b>14</b>
<b>Review of Related Literature and Studies</b>	<b>14</b>
2.1 Related Literature and Studies	14
2.1.1 Remote Sensing and Burn Severity Mapping	14
2.1.2 Google Earth Engine and Burn Severity Mapping	17
2.1.3 Web Mapping	20
<b>Chapter III</b>	<b>21</b>
<b>Methodology</b>	<b>21</b>
3.1 The United Nations' Recommended Practice for Burn Severity Mapping	21
3.1.1 The algorithm	21
3.1.2 Google Earth Engine code: the model	24
3.2 Frontend	26
3.3 Backend and server side application	28
3.4 Version control	30
3.5 Deployment	31
<b>Chapter IV</b>	<b>33</b>
<b>Results and Discussion</b>	<b>33</b>
4.1 Web App: GeeFire	33
4.2 Testing: Known bugs and defects	35
4.2.1 Error catching	35
4.2.1.1 MaxPixels limit	35
4.2.1.2 Pre-fire and Post-fire Date Ranges	36
4.2.1.3 Error 504 (HTTP 504 status code (Gateway Timeout))	36
Figure 9. Error 504 (HTTP 504 status code (Gateway Timeout))	37
4.2.2 Resetting the Map	37

<b>Chapter V</b>	<b>39</b>
<b>Summary, Conclusions, and Recommendations</b>	<b>39</b>
Summary	39
Conclusion	40
Recommendations	40
<b>References</b>	<b>42</b>

## List of Tables

Table 1: Burn severity classes and thresholds proposed by USGS. Color coding established by UN-SPIDER.

21

## **List of Figures**

Figure 1. Burn Severity Mapping Algorithm	22
Figure 2. GeeFire Frontend	27
Figure 3. GeeFire Github Repository	29
Figure 4. Vercel Main Page for GeeFire Web Application	30
Figure 5. GeeFire Frontend with Satellite Images, Classification, and Statistics	32
Figure 6. GeeFire Print Dialog Window	32
Figure 7. Vercel Logs	33
Figure 8. GeeFire Application and the Console logs (Error 500)	34
Figure 9. Error 504 (HTTP 504 status code (Gateway Timeout))	35
Figure 10. GeeFire application with console logs (Resetting the Map)	36

## **Abstract**

Wildfire is an unintentional fire burning in a natural area, usually a forest, often caused by human activity or by natural phenomena such as lightning. In the summer of 2019-20, Australia was hit by a severe bushfire, which led to an estimated death of 1.25 billion animals, up to 19 million hectares were burned, 12.6 million of which are forest and bushes, destroyed 3,094 homes, and killed 33 humans

There are existing methods for assessing the severity of forest fires, but there are currently no systems in place for calculating the post-fire devastation of an area in real time. The purpose of this research is to develop a web-based application to assess the severity of forest fires. The processing algorithm will be based on the United Nations' Recommended Practice for Burn Severity Mapping and utilize Google Earth Engine's readily accessible and ready-to-use data catalog. The NIR and SWIR of Sentinel 2 (Bands 8A and 12) or the NIR and SWIR2 of Landsat 8 (Bands 5 and 7) are recommended for calculating the Normalized Burn Ratio (NBR) to evaluate the severity of a fire.

This online application's main objective is to make it easier for stakeholders and laypeople to calculate post-fire damage without having to delve deeply into the subject's technical details.

# **Chapter I**

## **Introduction**

### **1.1 Introduction and General Background**

Forest fires have occurred naturally for millions of years, frequently resulting from lightning strikes or spontaneous combustion. (Bond et al., 2018) These types of fires contribute to the diversity of an ecosystem by exposing deceased trees and promoting new growth. However, the frequency and severity of wildfires have increased dramatically in recent years, with human activity playing a significant role in this trend.

Human activity, such as logging, urbanization, and agricultural expansion, has increased the amount of dry, flammable material in forests, making them more susceptible to wildfires. Human-caused flames, such as campfires, fireworks, and cigarettes, also contribute to the overall increase in forest fires.

Climate change has also had an impact on the frequency and severity of wildfires. Rising temperatures and changed patterns of rainfall have caused more droughts and increased the risk of wildfires (UNEP, 2022).

Wildfires have extensive impacts that encompass the natural ecosystem, wildlife, and human settlements. The wildlife may be left without food or shelter, and entire habitats may be destroyed by these fires. In addition, these activities have the potential to cause harm to watersheds, resulting in adverse effects such as soil erosion, water pollution, and compromised water quality. The loss of homes, businesses, and infrastructure also adversely affects human communities. Furthermore, the emission of smoke resulting from these fires can have adverse



effects on human health by aggravating respiratory conditions and inducing other health complications (UNEP, 2022).

In 2019, a bushfire in Australia had significant impacts on the environment and the population.

The fires had a devastating impact on the country's wildlife and natural habitat, with over 3 billion animals estimated to have been affected. Many unique species were pushed to the brink of extinction as their habitats were destroyed. The loss of biodiversity has long-term implications, as it affects the overall health of ecosystems and can lead to cascading effects through the food chain.

The bushfires also resulted in significant air pollution, with smoke and ash affecting air quality not just in Australia but also in neighboring countries such as New Zealand. The smoke caused health problems for many people, particularly those with respiratory issues.

Water contamination was also a major impact of the fires, as ash and other debris were carried by rainwater into rivers and other bodies of water. This contamination affected both aquatic life and human populations that relied on these water sources.

The fires also led to soil degradation, as the loss of vegetation left the soil exposed to erosion and nutrient loss. This degradation can have long-lasting effects on the soil's ability to support plant growth and can lead to further degradation of the surrounding ecosystem (UNEP, 2020).

As the frequency and severity of forest fires continue to increase, effective wildfire management and recovery efforts are essential. One critical component of these efforts is accurate and reliable assessments of burn severity, which can inform post-fire management and recovery efforts.

Burn severity mapping is a valuable tool for understanding the extent and severity of wildfire damage and helping land managers determine the best course of action for reestablishing vegetation, preventing soil erosion, and restoring ecosystem services.

One technology that has revolutionized remote sensing and shown promise in various applications is the Google Earth Engine. This cloud-based platform provides access to an extensive archive of satellite imagery and geospatial data, enabling the creation of detailed maps for various environmental applications. Google Earth Engine's powerful computing capabilities also allow for the processing and analysis of massive amounts of data, making it possible to produce accurate and reliable assessments of different environmental factors such as burn severity, flood simulation, drought monitoring, land cover change, and more (Gorelick et al., 2017).

Using Google Earth Engine's API, developers can create interactive web applications for mapping and visualizing various environmental phenomena. These web apps can help stakeholders and decision-makers better understand and manage different environmental challenges. With the ability to analyze and visualize vast amounts of data from space, Google Earth Engine is a powerful tool for understanding our planet and the complex interrelationships between different environmental factors (Gorelick et al., 2017).

## **1.2 Research Problem**

Wildfires have become a major environmental concern across the globe due to their devastating effects on human life, wildlife, and the environment. One critical aspect of wildfire management is the accurate and timely mapping of burn severity, which allows for effective decision-making regarding post-fire rehabilitation and recovery efforts. Currently, there is a lack of accessible and user-friendly web applications that utilize the United Nations recommended methodology for burn severity mapping. This has resulted in limited access to accurate burn severity maps by wildfire management teams, which can lead to inadequate post-fire rehabilitation efforts.

Therefore, the problem this thesis aims to address is the development of a web application that utilizes the United Nations recommended methodology for burn severity mapping using the Google Earth Engine platform, which offers a powerful computing capability that allows for the analysis of massive amounts of remote sensing data in a fast and efficient manner and will provide accessible and user-friendly burn severity maps for effective post-fire rehabilitation and recovery efforts. Further, it answers the following questions:

1. What is the United Nations recommended methodology for burn severity mapping and how does it compare to other pre-existing methodologies?
2. What are the technical requirements and considerations for developing a web application for burn severity mapping?
3. How can the web application be designed to ensure accessibility and user-friendliness for wildfire management teams?
4. How accurate and reliable are the burn severity maps produced by the web application, and how do they compare to other existing methods of burn severity mapping?
5. What are the potential challenges and limitations in implementing the web application in real-world wildfire management scenarios, and how can they be addressed?

## **1.3 Scope and Delimitation**

This study aims to develop a web application for burn severity mapping using the United Nations recommended methodology. The study will focus on utilizing remote sensing data from two of Google Earth Engine's data catalog (Sentinel-2 and Landsat 8) and its API for efficient data analysis and processing. The web application will be developed using open-source programming languages such as Python for the backend and JavaScript for the frontend to ensure its

accessibility and sustainability. However, this study will not include field validation of the burn severity maps generated by the web application. It will also not cover other methodologies for burn severity mapping aside from the United Nations recommended methodology. Additionally, the web application will be developed for use in areas where Landsat-8 and Sentinel-2 data are available and may not be applicable in areas with limited or no access to such data. Furthermore, in some cases, cloud cover may be so significant that it cannot be masked and mosaicked with available satellite images, resulting in the inability to produce an accurate classification. Finally, this study will cover the deployment of the web application, but the maintenance of the web application beyond the completion of the project may not be included.

## **1.4 Significance of the Study**

The proposed web application for burn severity mapping using the United Nations recommended methodology has significant implications for wildfire management and post-fire rehabilitation and recovery efforts. The web application will provide a user-friendly interface for generating accurate and reliable burn severity maps, which can aid in decision-making for resource allocation, risk assessment, and post-fire rehabilitation and recovery efforts. The use of the Google Earth Engine platform will allow for the efficient processing and analysis of remote sensing data, which can enable timely and accurate assessments of wildfire impacts on the landscape. The development and implementation of this web application can contribute to the advancement of wildfire management and post-fire rehabilitation and recovery efforts and provide a valuable tool for stakeholders involved in these efforts. Furthermore, this study can

serve as a reference for other researchers and practitioners interested in developing similar web applications for other environmental processes using remote sensing data.

## **Chapter II**

### **Review of Related Literature and Studies**

This chapter contains information from related literature and studies gathered from various references such as the internet, articles, theses, dissertations, and other printed or non-print reference materials which were found relevant in this study. The gathered information serves as supporting ideas in the conduct of this research.

#### **2.1 Related Literature and Studies**

##### **2.1.1 Remote Sensing and Burn Severity Mapping**

Satellite remote sensing refers to the technique of detecting and monitoring the physical attributes of a particular area by measuring its reflected and emitted radiation, usually from an aircraft or satellite, without physical contact. With the advancement of technology, acquiring high-resolution satellite images can now be used for different types of environmental research.

It involves the use of satellites equipped with sensors to collect data and images of the Earth from space. Remote sensing data can be used to study a wide range of environmental phenomena, including land use and land cover change, deforestation, urbanization, water quality, climate change, and forest fire. With advancements in satellite technology and remote sensing techniques, the capabilities of remote sensing have expanded and it has become a powerful tool for monitoring and assessing changes in the Earth's surface and atmosphere (Lillesand et.al., 2014).

Optical remote sensing involves the use of sensors that detect electromagnetic radiation within the visible, near-infrared, and thermal infrared parts of the electromagnetic spectrum. This type of remote sensing is particularly useful for studying the earth's surface, including vegetation cover, land use, and changes in water bodies. Optical sensors can be mounted on satellites, airplanes, and ground-based platforms (Jensen, 2016).

Radar remote sensing, on the other hand, uses sensors that emit microwave radiation and measure the time it takes for the radiation to reflect back from the surface. This type of remote sensing can penetrate through clouds, vegetation cover, and even some types of soil, allowing it to gather information about the earth's surface even in areas where optical sensors cannot be used. Radar sensors can also be mounted on satellites, airplanes, and ground-based platforms (Lillesand et al., 2015).

Given the potential broad spatial scales of wildfires, remote sensing technology offers an efficient means to examine the time-related features of a burned region (for instance, before, during, and after the fire occurrence) (Lentile et al. 2006). A reduction in chlorophyll in the burning region causes an increase in the visible region of the electromagnetic spectrum and a decrease in the near-infrared region (Escuin et al. 2008). Further, the electromagnetic spectrum's near-infrared (NIR) and short-wave infrared (SWIR) ranges (0.7 to 2.4  $\mu\text{m}$ ) are sensitive to post-fire affected vegetation, which can be utilized for burn severity mapping (Garcia and Caselles 1991).

A commonly used approach for determining the severity of a burn is by utilizing the Normalized Burn Ratio. This ratio is especially successful if applied to imagery acquired immediately before and after the burn (Key and Benson 2006). The first suggestion was to investigate the performance of using Landsat TM4 and TM7 bands for burn severity calculation, and results

have shown that NBR outperforms single-band Normalised Difference Vegetation Index (NDVI) (Key and Benson 1999; Epting et al. 2005). At the same time, the Differenced Normalised Burn Ratio (dNBR) was proposed, which uses the post-NBR and subtracts it from the pre-NBR which provides a more quantitative measure of the environmental differences caused by the fire (Key and Benson 1999).

The Normalized Burn Ratio, as defined by Key and Benson is computed using this formula:

$$NBR = \frac{\rho_{NIR} - \rho_{SWIR}}{\rho_{NIR} + \rho_{SWIR}}$$

where  $\rho$  is reflectance, and the subscripts NIR or SWIR describe the spectral regions while dNBR or Differenced Normalised Burn Ratio (dNBR) is defined as

$$dNBR = NBR_{prefire} - NBR_{postfire}$$

where the subscripts pre-fire and post-fire refer to the timing of the acquisition of the imagery.

dNBR has demonstrated remarkable efficiency in delineating the severity of a burn, particularly in forested areas (Wagtendonk, et.al, 2004). However, it is found to be less effective in other environments, such as grasslands (Lu et.al, 2016), and not optimal for mapping the burn severity in savannah regions (Roy et.al, 2006).

Landsat and Sentinel-2 are the medium resolution satellites that are most frequently utilized for mapping burn severity. The Monitoring Trends in Burn Severity (MTBS) Project, a collaboration between the USDA Forest Service and US Geological Survey, has been actively contributing to the development of Landsat data usage in monitoring wildfires on a national and global scale in recent times (Eidenshink et al., 2007). The project uses near-infrared and mid-infrared regions, which can be obtained from Landsat sensors (e.g., Landsat-5 TM and Landsat-7 ETM+ bands 4 and 7) to compute for NBR to map fire perimeters and severities (Key and Benson, 2006).



Another study focused on the use of recent Sentinel 2 satellite data to determine the extent of a forest fire and identify its severity in Çanakkale, Turkey, an area in the Mediterranean Basin prone to severe forest fires. The researchers utilized the dNBR index and pre- and post-fire Sentinel images to produce a map with a spatial resolution of 20 meters. The findings showed that the type of cover present affected the fire's severity, with forested areas suffering more serious effects due to their high levels of flammable fuel. Additionally, the study found that the use of Sentinel-2 data was a fast and precise method for monitoring forest fires due to its improved spatial and temporal resolution (Gökkaya, 2022).

### **2.1.2 Google Earth Engine and Burn Severity Mapping**

Google Earth Engine is an online platform that leverages Google's immense computing power to facilitate large-scale analysis of geospatial data related to crucial societal concerns such as deforestation, drought, disasters, disease outbreaks, food security, water management, climate monitoring, and environmental conservation (Gorelick et al., 2017).

Earth Engine has a readily available data catalog that has an intrinsic, high-speed computation service with multi-petabyte capacity. This service is accessible and modifiable via an Application Programming Interface (API) that is connected to an interactive development environment (IDE) available on the internet. This IDE allows for rapid prototyping and result visualization (Gorelick et al., 2017).

The data catalog has a broad collection of publicly accessible geospatial datasets, such as satellite and aerial images, weather and climate forecasts, environmental variables, socio-economic datasets, and topographic data. All this information is pre-processed into an easily accessible format that preserves the quality of the information and reduces the barriers associated with data management (Gorelick et. al., 2017).

Through a library of operators offered by the Earth Engine API, users can access and analyze both public and private data. These operators are implemented in a large-scale parallel processing system, that automatically divides and disseminates computations, thus providing high-throughput analysis capabilities. To access the API, users can use a thin client library or a web-based IDE that is built on top of the client library (Gorelick et. al., 2017).

Google Earth Engine is a crucial tool for mapping burn severity, as it allows researchers to access open datasets and powerful computational capabilities to calculate and map burn severity in real-time. Compared to other methods, which may require the download of satellite images and the use of QGIS or other remote sensing software, GEE is more convenient, as it only requires writing a single line of code in JavaScript to access an image collection. The only drawback of using Google Earth Engine for burn severity mapping is that users need to have some basic knowledge of JavaScript to use it effectively. However, there are numerous online resources available that utilize the Google Earth Engine code editor for analysis.

Parks et al. (2018) published a paper that discusses the use of Landsat-based fire severity datasets for research and monitoring purposes. The authors presented a new approach for producing three fire severity metrics delta normalized burn ratio (dNBR), the relativized delta normalized burn ratio (RdNBR), and the relativized burn ratio (RBR)—using the Google Earth Engine (GEE) platform. This approach involves a mean compositing method that eliminates the need for a priori scene selection, resulting in faster and easier production of fire severity datasets, and validated the GEE-derived fire severity metrics using field-based fire severity plots for 18 fires in the western United States and compared them to the standard approach that uses only one pre- and post-fire scene. The GEE-derived fire severity datasets showed better validation statistics compared to the standard approach. The authors provided code and a sample geospatial fire

history layer to produce these metrics and suggested that their GEE methodology can be applied to hundreds to thousands of fires, thus increasing the potential for fire severity monitoring and research globally. The study emphasized the importance of burn severity mapping and demonstrated the potential of GEE as an effective tool for producing such datasets.

Babu et. al (2020) utilized the Google Earth Engine cloud platform to generate burn severity maps for the eastern part of Australia using Landsat-8 and Sentinel-2 (S2) satellites. They compared severity maps of dNBR and RBR for both datasets based on the active fire product Visible Infrared Imaging Radiometer Suite (VIIRS) and found that RBR had higher accuracy than dNBR. The Sentinel 2-derived severity maps of both dNBR and RBR were more accurate than the Landsat 8-derived maps, attributed to the higher spatial resolution of Sentinel-2 datasets. The authors suggest that Sentinel-2 datasets can be a useful tool for rapidly generating burn severity maps using the Google Earth Engine platform.

The United Nations provides recommended practices for burn severity mapping to support disaster management and mitigation efforts. The recommended practice recommended the use of Landsat-8, Sentinel-2 and Google Earth Engine to produce burn severity maps. The practice outlines the use of the NBR using the Near-infrared(NIR) and Shortwave-infrared(SWIR) bands of the Electromagnetic Spectrum to calculate the Normalized Burn Ratio (NBR) for the pre- and post-fire scenarios. Delta NBR (dNBR) is then determined by the difference between the pre- and post-fire NBR. Finally, dNBR is classified according to the United States Geological Survey (USGS) standard for Burn Severity assessment.

The United Nations emphasizes, however, that the software used affects the accuracy of area measurements. When comparing the application of Landsat burn severity methodology in both QGIS and Google Earth Engine, the resulting maps appear identical visually. However, the

calculation of the area for each class demonstrated a variation of up to 1% of the total study area. This divergence in reported hectares is attributed to the projection assumptions used in the procedures.

### **2.1.3 Web Mapping**

Many repositories can be found on Github that utilize the Google Earth Engine API for burn severity mapping. The majority of these are not deployed on the web, and can only be accessed within the Google Earth Engine environment. However, it is worth noting that there are limited applications deployed on the web that are limited to a single area of interest. One example of this is IndigoWizard's application named "wildfire-burn-severity" which aims to map the burn severity of the wildfire that occurred in Mount Chenoua, Tipaza, Algeria on August 14th-16th 2022 (Mokhtari, 2022).

However, Cesar Aybar created a web application that uses Google Earth Engine to implement the United Nations' suggested method for flood detection into practice (Aybar, 2021). The frontend of the application was developed using JavaScript, HTML, and CSS, while the backend was created using FastAPI.

## **Chapter III**

### **Methodology**

This chapter provides details about the research procedures, sources of data, data gathering procedures, instrumentation, and statistical treatment of the study. Discussion of the entire research design is the main focus in this chapter.

#### **3.1 The United Nations' Recommended Practice for Burn Severity**

##### **Mapping**

##### **3.1.1 The algorithm**

The algorithm as defined by the United Nations Office for Outer Space Affairs aims to create efficient yet reliable burn severity maps for any area of interest. While there are several approaches for burn severity mapping, Normalized Burn Ratio is utilized as it was designed to highlight burned areas and estimate burn severity. It uses near-infrared (NIR) and shortwave-infrared (SWIR) wavelengths. Before the fire, healthy plants had a very high NIR reflectance and a low SWIR response. On the other hand, newly burnt lands have a low NIR reflectance and a high SWIR reflectance. The NBR is calculated for pre-fire and post-fire images, and the difference between the two (delta NBR or dNBR) is used to assess burn severity. High dNBR values indicate more severe damage, while negative dNBR values may signify increased vegetation productivity. The severity of burn can be classified using the ranges suggested by the United States Geological Survey (USGS)(Key and Benson, 2006).

Severity Level	dNBR Range (scaled by $10^3$ )	dNBR Range (not scaled)
Enhanced Regrowth, high (post-fire)	-500 to -251	-0.500 to -0.251
Enhanced Regrowth, low (post-fire)	-250 to -101	-0.250 to -0.101
Unburned	-100 to +99	-0.100 to +0.99
Low Severity	+100 to +269	+0.100 to +0.269
Moderate-low Severity	+270 to +439	+0.270 to +0.439
Moderate-high Severity	+440 to +659	+0.440 to +0.659
High Severity	+660 to +1300	+0.660 to +1.300

Table 1: Burn severity classes and thresholds proposed by USGS. Color coding established by UN-SPIDER.

The process starts with specifying the pre-fire and post-fire range dates. Knowing the pre and post-fire range dates is important in burn severity mapping because it allows for an accurate assessment of the severity of the fire and the resulting changes in the landscape.

Pre-fire imagery offers a baseline for the region's pre-fire conditions, which is crucial for accurately estimating the burn's extent and identifying areas that were more or less severely affected by the fire. The post-fire imagery allows for the identification of changes in the landscape following the fire, including changes in vegetation cover, soil erosion, and other impacts.

It then uses the NIR and SWIR bands of the electromagnetic spectrum, and the image collection of the range dates is atmospherically corrected before computing for Normalized Burn Ratio. The pre-fire NBR and post-fire NBR are used to calculate for dNBR, and it uses this to classify the image using the burn severity classes and thresholds proposed by USGS.

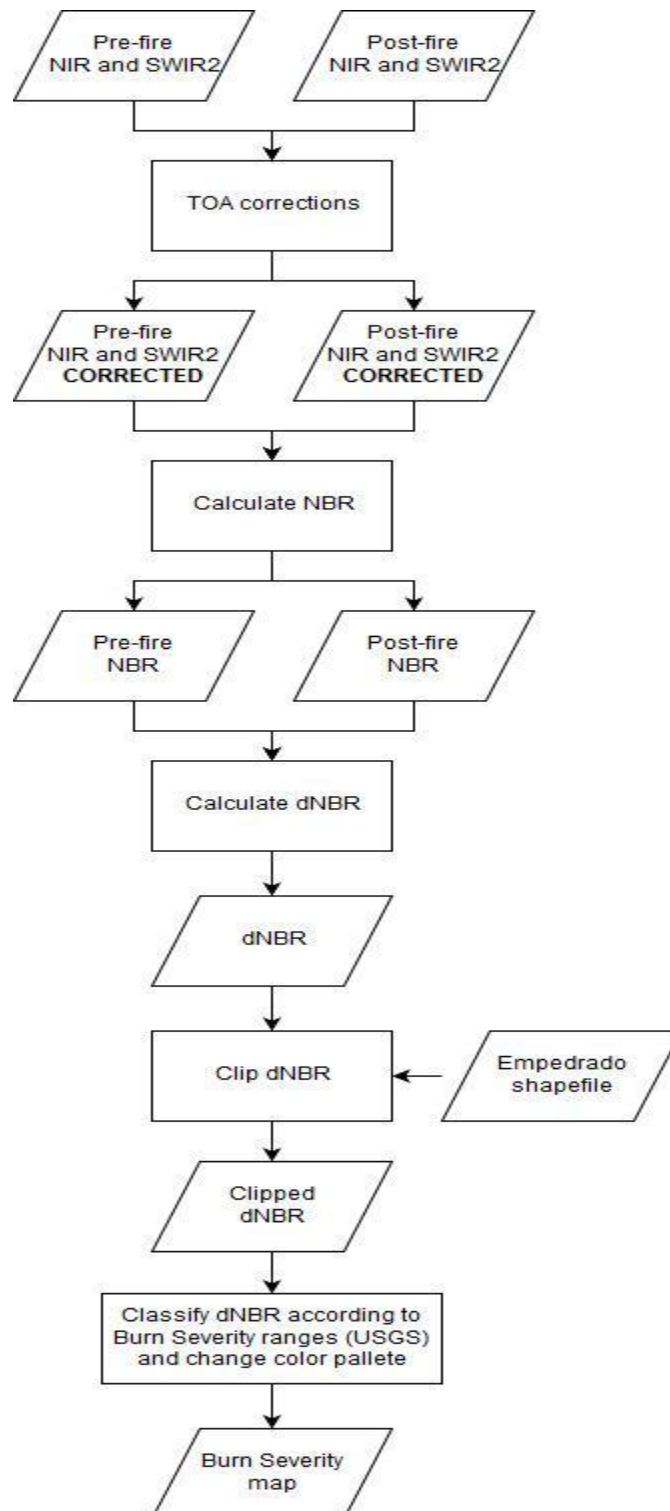


Figure 1. Burn Severity Mapping Algorithm

### 3.1.2 Google Earth Engine code: the model

Originally, the code was written in Javascript. To make the code reusable, the model or the logic of the application is rewritten in Python and then used in functional programming.

The model is divided into several functions: the first one is preprocessing, which is used to preprocess the satellite images based on the parameters set by the user in the frontend. It takes four input parameters: *"ee\_geom"* is a geographic location of interest defined as a geojson object; *"satellite"* is a string indicating which satellite data to process (*"Sentinel-2"* or *"Landsat-8"*); *"preFire\_period"* and *"postFire\_period"* are date ranges that specify the time period before and after a wildfire event, respectively.

Based on the input *"satellite"* value, the function processes the Sentinel-2 or Landsat-8 satellite imagery collections from the Google Earth Engine (GEE) platform. The pre- and post-fire imagery collections are filtered by the date ranges and location, and the cloud masking algorithm is applied to obtain cloud-free imagery.

The function then creates mosaicked true color images and cloud-masked images for the pre- and post-fire periods, and calculates the Normalized Burn Ratio (NBR) for each period. The difference between the pre- and post-fire NBRs is calculated to derive the delta NBR or dNBR, which is then scaled to USGS standards. The resulting dNBR image is then classified into eight burn severity classes based on the provided threshold values.

Finally, the function returns a dictionary of Earth Engine objects containing the pre- and post-fire mosaicked true color images, the cloud-masked images, and the dNBR image classified into eight burn severity classes.



The values returned from this function will then be used for the next function, `"burnSeverity"` which is used to calculate for the statistics. The `"burnSeverity"` function takes in a dictionary, `"pre_processing_params"`. It extracts three variables from this dictionary: `"classified"`, `"area_of_interest"`, and `"satellite"`.

The function then checks the value of the satellite. If it is "Sentinel-2", the function calculates the total number of pixels in the classified layer within the `"area_of_interest"` geometry. It then defines a nested function, `areacount`, which calculates the number of pixels, hectares, and percentage of total area for each class within the classified layer. For each class, the `areacount` function creates a mask, counts the number of pixels in that class within the `"area_of_interest"`, and calculates the corresponding hectares and percentage of the total area. The results for each class are then appended to the `arealist` list.

If the satellite is not "Sentinel-2", the function performs similar calculations but using different scaling factors to account for the different pixel size of the satellite imagery. Finally, the function returns the `arealist` list which contains the area statistics for each class within the classified layer.

The third function is the masking function, which is used for the preprocessing of the satellite image collection. The functions depend on the type of satellite used: Sentinel-2 and Landsat-8. The `"maskS2sr()"` function masks out cloudy and cirrus pixels in Sentinel-2 images by creating a binary mask where all pixels with values of 10 and 11 are set to zero. These values represent clouds and cirrus, respectively. The function first extracts the Quality Assessment (QA) band from the input image, applies the binary mask to it using the `.bitwiseAnd()` method, and checks if the resulting mask is equal to zero. The function then applies the

resulting mask to the input image using the `.updateMask()` method and copies the start time of the input image to the output image properties.

The `"maskL8sr()"` function masks out cloud, cloud shadow, and snow pixels in Landsat-8 images by creating a binary mask where all pixels with values of 3, 4, and 5 are set to zero. These values represent cloud shadows, snow, and clouds, respectively. The function first extracts the pixel QA band from the input image, applies the binary mask to it using the `.bitwiseAnd()` method, and checks if the resulting mask is equal to zero. The function then applies the resulting mask to the input image using the `.updateMask()` method, selects all bands with names starting with "B" using the `.select()` method, and copies the start time of the input image to the output image properties.

And finally, the last function is `"display_map"` which takes the `"pre_processing_params"` as the parameters. The function uses the input parameter to display a map with different layers depending on the satellite that was used to capture the images.

The function visualizes the pre-fire and post-fire true color mosaicked images, the pre-fire and post-fire cloud-masked mosaicked images, a gray-scale image of the dNBR (delta Normalized Burn Ratio), and a classified image. It then returns a dictionary called `display_layer` containing the tile ID of each of these layers, which will send a request to Google Earth Engine for visualization in the app.

## 3.2 Frontend

The frontend of a website refers to the part that the user interacts with. It's the visible part of the website that includes the layout, design, and functionality. HTML and CSS are two essential

building blocks of the frontend. HTML, which stands for Hypertext Markup Language, is used to structure content on the web. HTML elements are used to define the structure of a web page, such as headings, paragraphs, lists, and links. CSS (Cascading Style Sheets) is used to define the visual appearance of the HTML elements, such as the font styles, colors, and layout.

In the codebase, map.html is an HTML file that creates a map using the OpenLayers JavaScript library. The HTML code in map.html defines the structure of the map and includes elements such as a map container, tile layers, and popups. The CSS stylesheets are used to define the visual appearance of the map, such as the background color, font styles, and marker icons. The JavaScript files contain code that adds interactivity to the map, such as zooming in and out, displaying the tiles (satellite images and classification), a legend, and a statistics table. Meanwhile, OpenLayers was used to create a base map, and the ol-layerswitcher module (<https://github.com/walkermatt/ol-layerswitcher>) to enable and disable layers.

The UI features a sidebar that includes several tabs for setting pre- and post-fire dates, selecting imagery, visualizing maps, displaying statistics, downloading maps and statistics, and resetting the form. Additionally, users are expected to create a rectangle on the map to serve as the area of interest for the fire area.

Once the user has selected the parameters, drawn an area of interest, and sent a request to the backend, a map will be sent to the frontend that shows the pre- and post-fire mosaiced satellite images and the classification image tiles. The statistics table is also available for display at the user's request. Another function allows the user to download the map in PDF or PNG format.

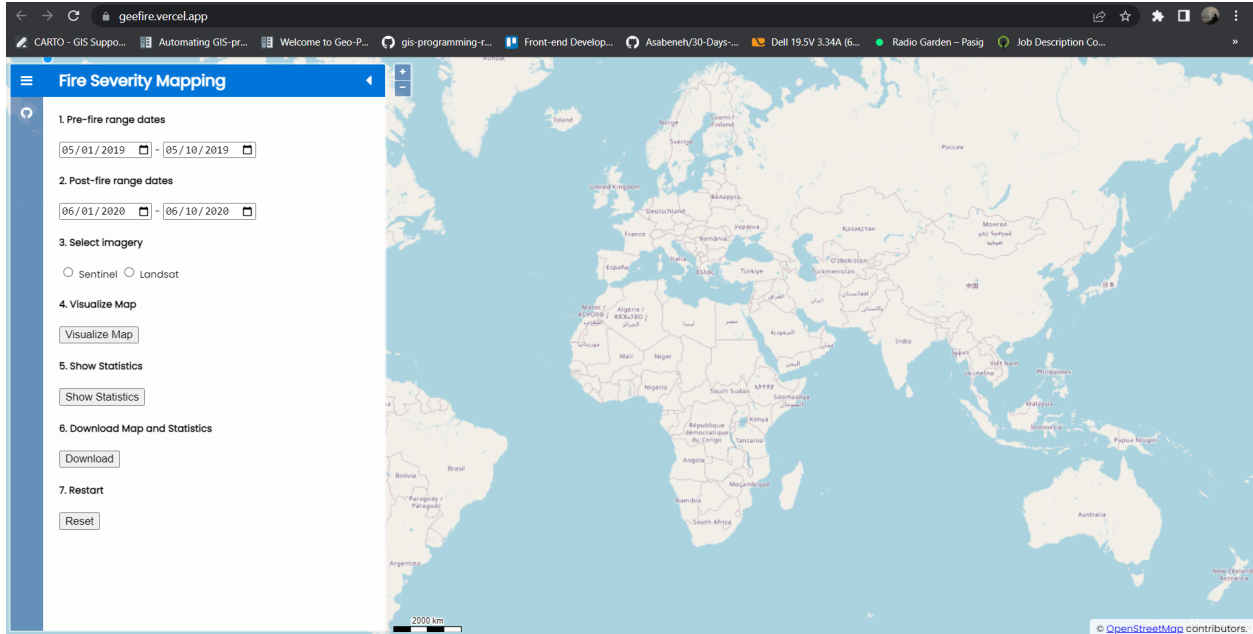


Figure 2. GeeFire Frontend

### 3.3 Backend and server side application

Flask is a lightweight and flexible web framework that allows developers to quickly build web applications using Python. It is built on top of the Werkzeug toolkit and the Jinja2 templating engine, providing a simple and intuitive way to handle HTTP requests and responses (Flask, n.d.). Flask uses *decorators* to define routes for different endpoints and provides a built-in web server that can be used for development and testing. The `app.py` file in the GeeFire repository showcases how Flask can be used to build a web application that provides a graphical user interface for running a burn severity analysis using Google Earth Engine. By defining routes for different endpoints, Flask allows users to interact with the web application by sending HTTP requests with different parameters. The `app.py` file contains the server-side backend code for the GeeFire application. It is responsible for processing user requests, running the appropriate Earth Engine processing functions, and returning the results to the client.

The first few lines of the code in `app.py` define the required imports, including Flask, Earth Engine, and the custom functions defined in the `src` directory. The `"before_request"` decorator is used to initialize the Earth Engine API with the user's credentials before processing any requests.

The `@app.get('/')` decorator is used to define a route for the default homepage. When a user visits the website, they will be directed to the `map.html` file, which contains the client-side code for displaying the map and interacting with the backend.

The `@app.route('/visualize', methods=['POST'])` decorator is used to define a route for the visualize endpoint, which is responsible for processing a user's request to generate a burn severity map. When a user submits the required parameters (bounding box coordinates, date range, and satellite data type), the request is processed using the `preprocessing()` and `display_map()` functions defined in the `src.model` module. The output is then returned to the client as a tile ID, which is used to display the map on the frontend.

The `@app.route('/statistics', methods=['POST'])` decorator is used to define a route for the statistics endpoint, which is responsible for processing a user's request to generate burn severity statistics for a given area. When a user submits the required parameters, the request is processed using the `preprocessing()` and `burnSeverity()` functions defined in the `src.model` module. The output is then returned to the client as a JSON object containing the statistics.

The `@app.route('/download', methods=['POST'])` decorator is used to define a route for the download endpoint, which downloads the map in different formats.

### 3.4 Version control

Visual Studio Code, a popular code editor that offers a range of features for writing and debugging code, was used to write the whole codebase. One of the reasons Visual Studio Code was chosen as the IDE is its versatility, as it supports multiple programming languages and has a vast library of extensions that can be used to enhance the coding experience. It has a user-friendly interface, which makes it easy to navigate and customize to the user's needs.

To manage the codebase and keep track of changes over time, Git was used for version control. Git is a widely used distributed version control system that allows multiple people to collaborate on the same codebase without overwriting each other's work. With Git, a repository for the codebase could be created, enabling it to track changes, revert to previous versions, and merge code from different branches. This gave it more control over the development process and helped avoid errors and conflicts.

Finally, the codebase was uploaded to Github, a popular web-based hosting service for Git repositories. By doing so, the codebase can be shared with others, collaborated with by other developers, and contributed to open-source projects. GitHub also provides tools for issue tracking, code review, and continuous integration, making it a valuable platform for software development. The codebase is available on GitHub: <https://github.com/ruthie02/GeeFire>

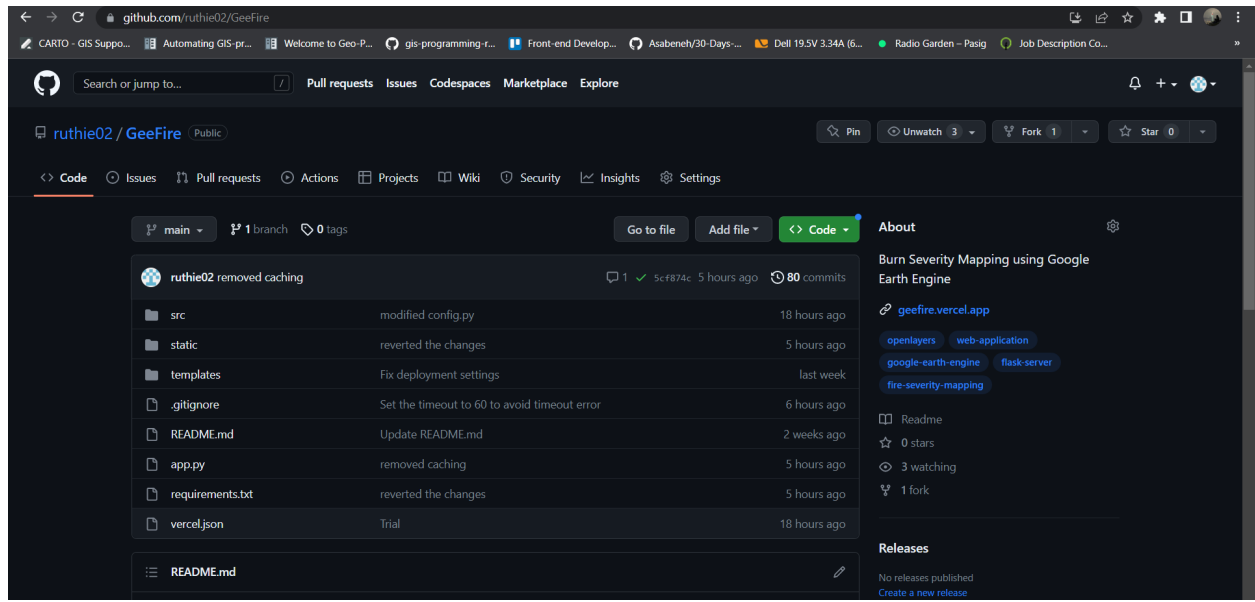


Figure 3. GeeFire Github Repository

## 3.5 Deployment

Vercel is a cloud platform that provides a range of services for building, deploying, and scaling web applications (Vercel, 2023). It's built around the concept of serverless computing, which allows developers to write code that runs on demand without needing to manage servers or infrastructure.

To deploy an app using Vercel's serverless functions, the user first needs to create an account on their website. Once the user has an account, a new project can be created and connected to a Git repository. Once deployed, Vercel will automatically detect the new code and deploy it to their servers.

GeeFire was deployed using a serverless function that is written in Python. The `vercel.json` file, which Vercel uses as a configuration file to define various settings for a project's deployment, and `requirements.txt`, which installs the packages and libraries used in the app, are both

necessary. This includes the project's name, the build command used to compile the code, environment variables, and the routes used by the application.

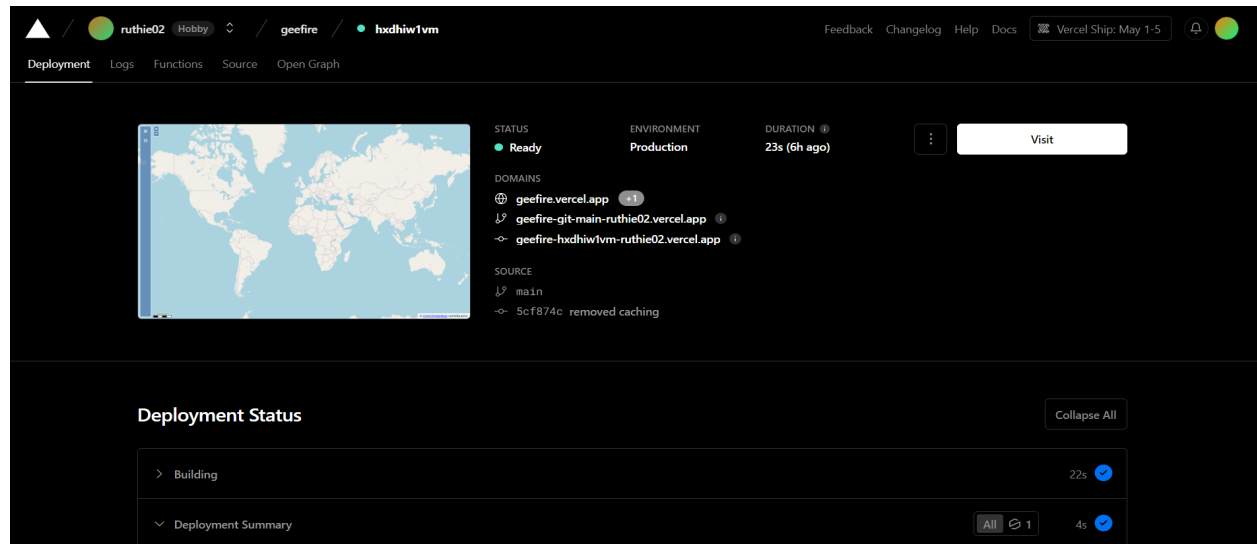


Figure 4. Vercel Main Page for GeeFire Web Application



## **Chapter IV**

### **Results and Discussion**

This chapter discusses the results of the web application created.

#### **4.1 Web App: GeeFire**

The final product is a web application called GeeFire. It is accessible to anyone with internet access. This link will take you to the website: <https://geefire.vercel.app/>.

The user is able to explore a map in Web Mercator projection (OpenLayers' default) and a dockable sidebar. The user must set the pre- and post-fire range dates, as well as the satellite image type (Landsat-8 or Sentinel-2) in the sidebar. By choosing the visualize map button, a request is sent to the API, which returns the tiles of the mosaiced images: True Color and Cloud Masked satellite images, dNBR in gray, and dNBR classified. These layers can be disabled or enabled using the layer switcher control, and the legend is displayed in the application's lower left corner.

When the Show Statistics button is selected, an API request is sent to calculate the statistics. Once the computation is complete, a table containing the Class Name, Percentage, Hectares, and Pixel values will appear in the bottom-middle portion of the map.

The next function is the Download button. After clicking the button, a dialog box will appear requesting the user's preferences prior to saving the file as a jpeg, png, or pdf.

Lastly, by selecting the Reset button, the user can delete all content from the map.

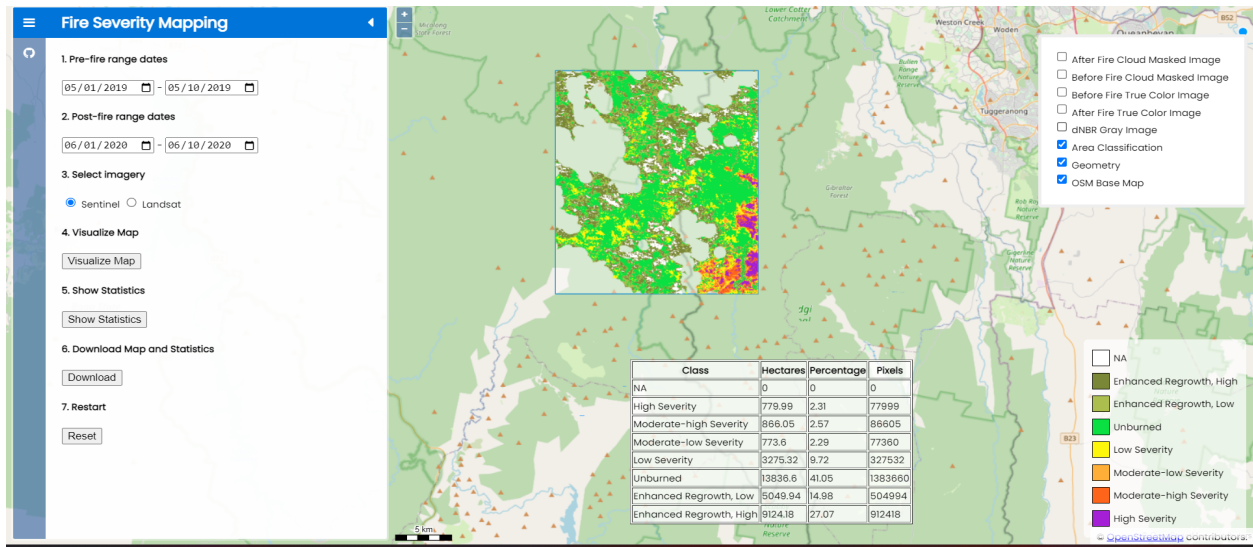


Figure 5. GeeFire Frontend with Satellite Images, Classification, and Statistics

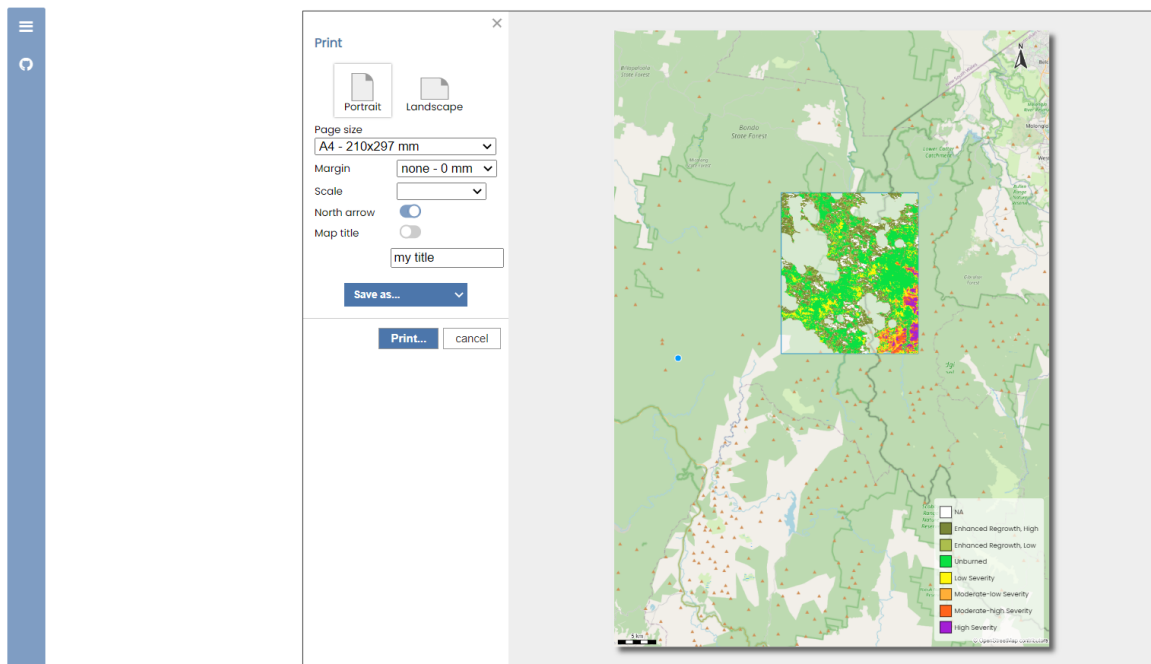


Figure 6. GeeFire Print Dialog Window

## 4.2 Testing: Known bugs and defects

### 4.2.1 Error catching

#### 4.2.1.1 MaxPixels limit

Catching errors is excluded from the first part of development. The main goal in the early stages of web development is to build a usable and as accurate as possible application that is available online. Therefore, when a user creates an area of interest that is huge, the app might be able to load the tiles of the mosaiced satellite images and the classification image but will not show any error in the frontend that the server couldn't compute for the statistics. Unless the user checks the console to look for issues, the user may never be able to know that the process has stopped executing because of the maximum limit of pixels allowed by the Google Earth Engine API for the function *Image.reduceRegion*.

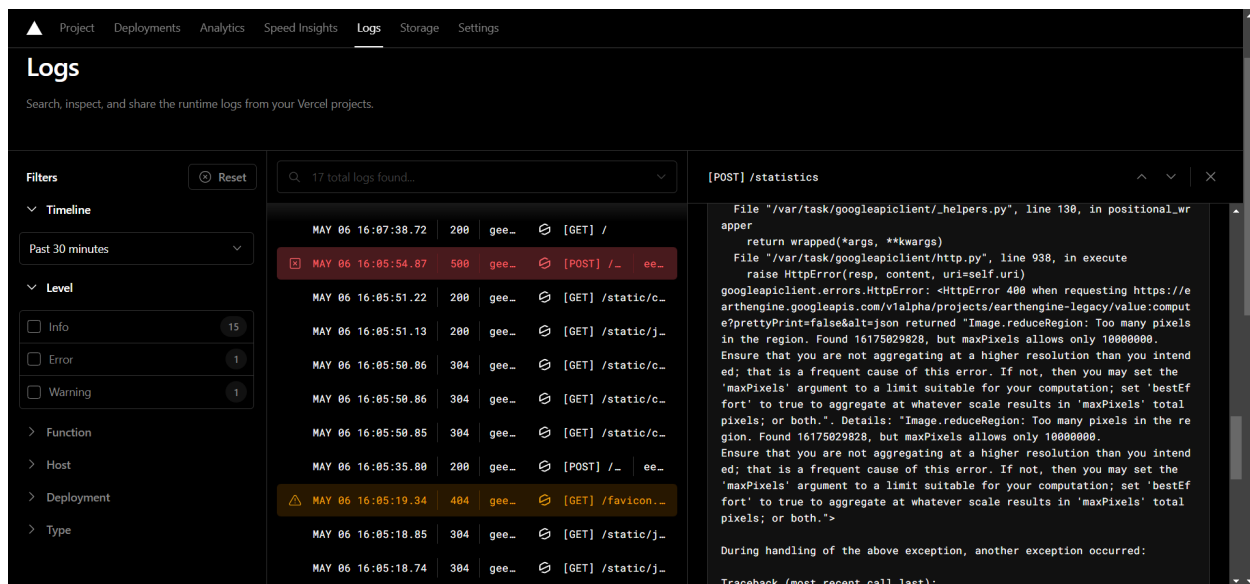


Figure 7. Vercel Logs

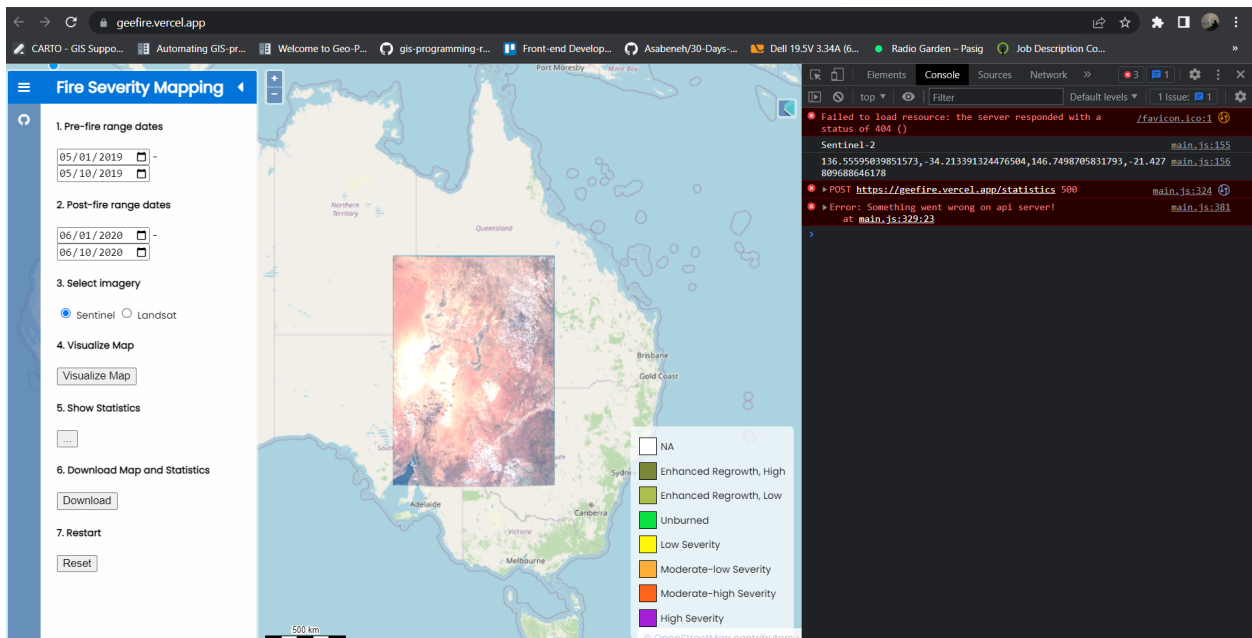


Figure 8. GeeFire Application and the Console logs (Error 500)

#### 4.2.1.2 Pre-fire and Post-fire Date Ranges

Although highly unlikely, in the worst-case scenario, a user might choose dates that crash the server. For example, choosing range dates that are before the Sentinel-2 or Landsat-8 were launched in space or beyond the current date would result in no satellite images being retrieved from the GEE API. Further, if a user chooses a long-range date, the server might crash when trying to mosaic images of multiple dates. The user will never be notified that the error has occurred unless the console is checked for possible errors.

#### 4.2.1.3 Error 504 (HTTP 504 status code (Gateway Timeout))

The whole codebase isn't optimized, so every function takes a long time to execute. Vercel's free tier has a limit of 10 seconds for execution time. Using Google Earth Engine's capability to

compute the statistics takes a bit of memory and time; therefore, Vercel is stopping the execution and returning error 504.

If a user wants to compute the statistics, it is recommended to run the app locally by cloning the repository and registering with Google Earth Engine to get access to the API.

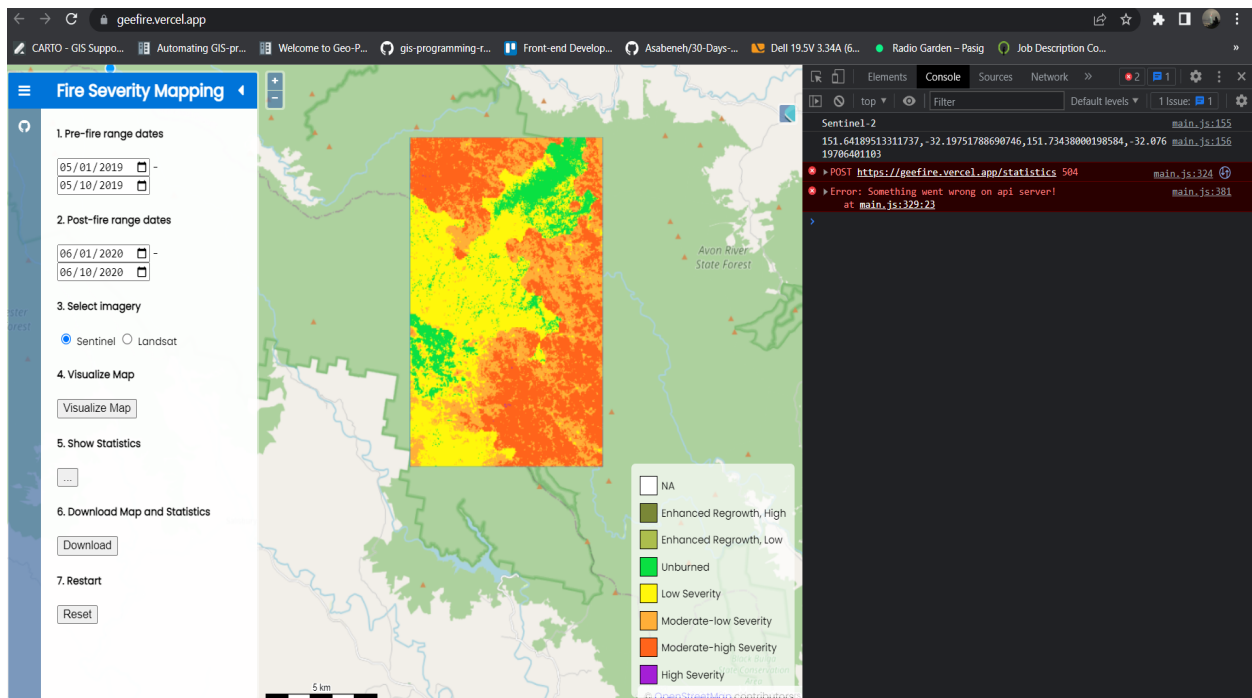


Figure 9. Error 504 (HTTP 504 status code (Gateway Timeout))

### 4.2.2 Resetting the Map

Resetting the map works but won't remove the legend once the Visualize function is executed. Instead of resetting, it is recommended to refresh the website and supply the required parameters again.

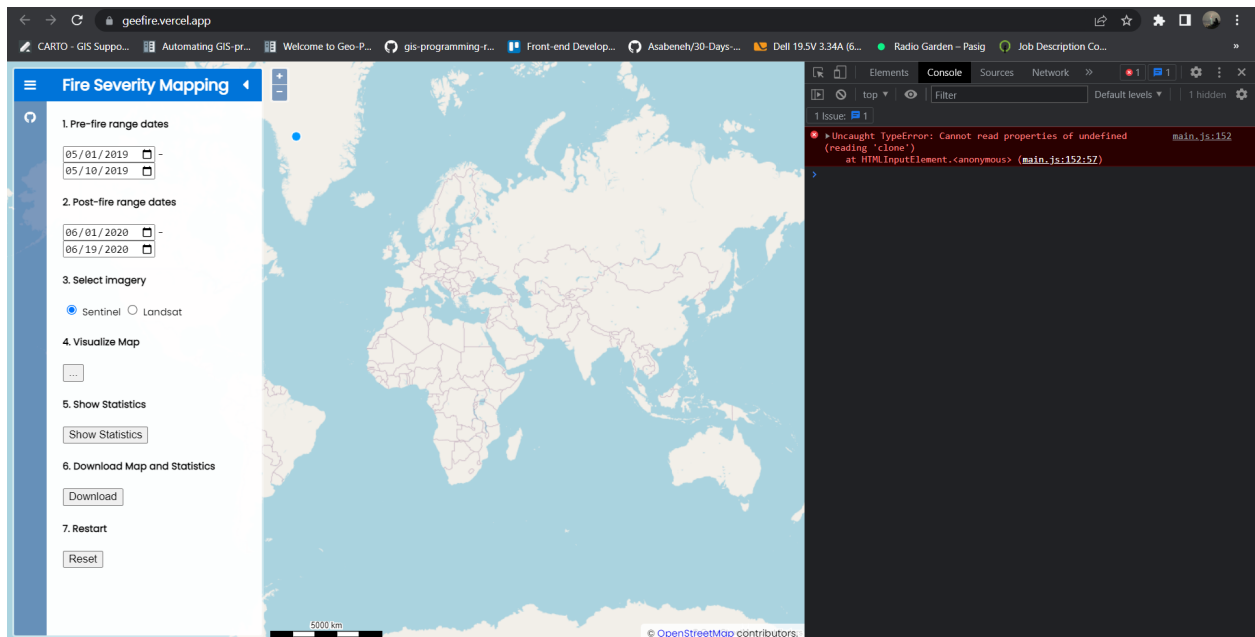


Figure 10. GeeFire application with console logs (Resetting the Map)

## **Chapter V**

### **Summary, Conclusions, and Recommendations**

Presented in this chapter are the summary of the study and findings, as well as conclusions and recommendations.

#### **Summary**

In this research, a website was created to map and evaluate the severity of burn using UN-Spider's recommended practice. The project was completed using the computing power and vast collection of satellite images available through the Google Earth Engine API. The Normalized Burn Ratio (NBR) was calculated using the NIR and SWIR bands of Sentinel 2 (Bands 8A and 12) or the NIR and SWIR bands of Landsat 8 (Bands 5 and 7) to determine the severity of the burn. Python and Flask were used for the backend and server-side applications, while Javascript, HTML, and CSS were used for the frontend. Version control was maintained using Git and GitHub, and the project was deployed using Vercel.

The web app was tested for bugs and defects, and mostly it concerns error catching and optimization (performance). The app in production cannot compute statistics as the free tier of Vercel for serverless applications limits the execution time to 10 seconds. The function to compute the statistics is taking more than a minute on average, thus returning a gateway timeout response.

All of the bugs caught can be solved by optimizing the code, but for the first phase of development, optimization wasn't the priority.

If users want to compute the statistics, they can clone the repository onto their local machine, register for an account in Google Earth Engine, and do the computation themselves. However, if users only want the map, they can use the deployed web app.

## **Conclusion**

Google Earth Engine's powerful computational capability and freely available, ready-to-use data catalog are huge milestones in the geospatial industry. Any scientist with a basic knowledge of programming can do geospatial analysis on-the-fly. Google Earth Engine has a built-in web coding editor where the user can do analysis and visualization in the browser. With this, the user doesn't need to install any GIS software on their computer as they only need a browser, and the computation is done in the cloud, therefore not requiring too much computer memory.

By taking advantage of this, Google Earth Engine can be a huge help not only to GI scientists but also to stakeholders who need rapid assessment of any disasters. With the right skills and knowledge, any natural phenomenon can be assessed, computed, and visualized easily, and it can even be published on the web for anyone to use.

## **Recommendations**

System design is important when building a web application. Although the computation is done through Google's cloud, the communication between the servers can take a bit of time, which may result in timeouts. Therefore, choosing the right data structures can help with optimization.

Further, knowledge of deployments and system infrastructure is recommended to ensure that the web app is seamless and works correctly. Running the application on the local computer versus



on the web can be different in certain scenarios; therefore, careful planning on how to deploy the web app and what tools to use is recommended.

## References

- Aybar, C. (2021). ee-fastapi [GitHub repository]. <https://github.com/csaybar/ee-fastapi>
- Bond, W. J., & van Wilgen, B. W. (2012). Fire and plants. Springer Science & Business Media.
- Babu, K. V. S., & Vanama, V. S. K. (2020). Burn area mapping in Google Earth Engine (GEE) cloud platform: 2019 forest fires in eastern Australia.
- Eidenshink, J.C.; Schwind, B.; Brewer, K.; Zhu, Z.-L.; Quayle, B.; Howard, S.M. (2007). A project for monitoring trends in burn severity. *Fire Ecol.*
- Epting, J., Verbyla D., & Sorbel, B. (2005). Evaluation of remotely sensed indices for assessing burn severity in interior Alaska using Landsat TM and ETM+.
- Escuin, S., Navarro R., & P. Fernández. (2008). Fire severity assessment by using NBR (Normalized Burn Ratio) and NDVI (Normalized Difference Vegetation Index) derived from Landsat TM/ETM images.
- García, M.J.L. & Caselles, V. (1991). Mapping burns and natural reforestation using thematic mapper data.
- Gökkaya, K. (2021). Burned Area and Fire Severity Prediction of a Forest Fire Using a Sentinel 2-Derived Spectral Index in Çanakkale, Turkey.
- Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., & Moore, R. (2017). Google Earth Engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment*.
- Jensen, J. R. (2016). *Remote Sensing of the Environment: An Earth Resource Perspective*. Pearson Education.
- Key, C.H.; Benson, N.C. (1999). Measuring and remote sensing of burn severity. *Proceedings joint fire science conference and workshop*, vol. 2. University of Idaho and International Association of Wildland Fire Moscow, ID, p. 284.

- Key, C.H.; Benson, N.C. (2006). Landscape assessment (LA). In FIREMON: Fire Effects Monitoring and Inventory System; General Technical Report RMRS-GTR-164-CD; U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station: Fort Collins, CO, USA.
- Lentile, L. B., Holden, Z. A., Smith, A. M. S., Falkowski, M. J., Hudak, A. T., Morgan, P., ... & Gessler, P. E. (2006). Remote sensing techniques to assess active fire characteristics and post-fire effects.
- Lillesand, T. M., Kiefer, R. W., & Chipman, J. W. (2014). Remote sensing and image interpretation. John Wiley & Sons.
- Lu, B., He, Y., & Tong, A. (2015). Evaluation of spectral indices for estimating burn severity in semiarid grasslands.
- Mokhtari, A. I. (2022). Wildfire-burn-severity [GitHub repository].  
<https://github.com/IndigoWizard/wildfire-burn-severity/>
- Pallets. (2023). Flask Documentation (Version 2.3.x). Retrieved May 13, 2023, from <https://flask.palletsprojects.com/en/2.3.x/>
- Parks, S.A., Holsinger, L.M., Voss, M.A., Loehman, R.A. and Robinson, N.P. (2018). Mean Composite Fire Severity Metrics Computed with Google Earth Engine Offer Improved Accuracy and Expanded Mapping Potential.
- Suryadi, F. X., & Wibowo, W. A. (2019). Burned area mapping using Google Earth Engine: a case study of peatland fires in Riau, Indonesia.
- Tanase, M. A., Calugaru, V., & Ioja, I. C. (2020). Assessing the Performance of MODIS and Sentinel-2 Burn Severity Indices for Fire-Scar Mapping in Romania.

Tian, Y., Brandt, J. S., Radeloff, V. C., Chen, Y., Li, G., & He, Y. (2019). Burn severity maps from Landsat predict forest microclimate two years after mixed severity wildfire.

Vercel. (n.d.). Vercel Documentation. Retrieved May 9, 2023, from <https://vercel.com/docs>

Wang, W., Li, X., Li, Y., & Guo, Q. (2019). Multi-source remote sensing for mapping burn severity of fire events: a review.

Woldai, T., Valeo, C., & Pham, B. (2019). Mapping burn severity using Landsat 8 OLI and TIRS in a temperate forest landscape of western Canada.

Yang, X., Cheng, J., Lu, J., Chen, Y., & Liu, Y. (2019). Mapping burn severity using Landsat and Sentinel-2 data in combination with a random forest model.

Zhao, F., Ramoelo, A., Skidmore, A. K., Wang, T., & Darvishzadeh, R. (2018). Multi-source remote sensing data fusion for mapping post-fire burn severity in a Mediterranean oak woodland.

## DECLARATION

I, the undersigned, Bongon Mary Ruth (NEPTUN CODE: XFOUAQ), declare that the present master's thesis is my original intellectual product in full and that I have not submitted any part or the whole of this work to any other institution. Permissions related to the use of copyrighted sources in this work are attached.

I agree to the publication of the accepted master's thesis in pdf form on the website of the Department of Cartography and Geoinformatics.

Budapest, 13th of May, 2023



---

(signature of the student)