M2-TSI UE52 : EARTH OBSERVATION

STUDY OF THE 2020 CALIFORNIA WILDFIRES' IMPACT ON THE ATMOSPHERIC CARBON MONOXIDE

Enguerran VIDAL - Javier SOTO-AGUILAR - Théau CHIAPELLA

April 1, 2022

Abstract

Earth's surface hosts extensive amounts of human related processes that have an impact on the atmosphere and carry an interest on a global scale. This report contains a deep analysis discussing the specific relation between the wildfires seen on the continental United States' West Coast and the carbon monoxide levels observed in the atmosphere above. We accomplished our goals through the extraction and processing of data-sets of said geographical area from the MOPITT and MODIS instruments on-board the current TERRA mission from NASA. The primary results of our processing show a clear correlation between the fire and the Carbon Monoxide abundance distribution in the air. A further analysis shows a 6 times increase of wildfire surface in the western part of the country between the 2015 and 2020 Western United States Summer wildfire seasons, while the total Carbon Monoxide emissions are in decline, concluding to the fact that the overall impact of these wildfires grows larger in our changing climate.

Keywords: Carbon Monoxide, California, Wildfires, MOPITT, MODIS

Contents

Chapter 1 Introduction **3**

Chapter 2 Methods and Proceidures **4**

2.1. Data-sets and Study Extents 4

2.1.1. Carbon Monoxide Levels from MOPITT 4

2.1.2. Active Fires Locations from MODIS 4

2.1.3. Study Extents 5

2.2. Data Extraction and Processing 7

2.2.1. Extracting data and averaging for Imaging 7

2.2.2. Plotting with Cartopy 8

Chapter 3 Results **9**

3.1. Study of the amount of wildfires 9

3.1.1. Initial Wildfires surface area Validation 9

3.1.2. Comparison of the amounts of wildfires in time-series 9

3.2. Study and Comparison of the Carbon Monoxide Levels 10

3.3. Correlation between Wildfires and Atmospheric CO Levels 11

Chapter 4 Conclusion **13**

Chapter A Codes **15**

Introduction

Pine forests in Mediterranean climate zones tend to be subject to bursts of wildfires in a periodic basis. As a part of the natural cycle of these forests, the tree renewal process takes into account this fires to the point of relying on this phenomenon to regenerate themselves, killing larger trees often to make way for new generations of pines. This is especially true in the state of California where these wildfires happen regularly each year. However, due to anthropogenic climate change, the severity, frequency, and total area burnt have heavily increased year by year, leading to disastrous environmental consequences. In the last decade, there have been several years with exceptionally large fires. Earlier records show fires of similar size in the nineteenth and early twentieth century. Lengthy droughts, as measured by the Palmer Drought Severity Index (PDSI), were associated with the peaks in large fires in both the 1920s and the early twenty-first century. This culminated last year in the summer of 2020 when the United States of America western coast saw its most extreme wildfires season on record, with over 41000 square kilometers of burnt land from hundreds of wildfire starts. These wildfires can be tracked through the study of carbon monoxide levels above ground coming from the incomplete combustion of shrubs, dry leaves, and wood, and found to have to a greater contribution to the overall CO levels above USA mainland in summer of 2020. This project has the intent to analyse the CO contributions of the 2020 West Coast wildfires in the scale of the entire US mainland CO emissions.

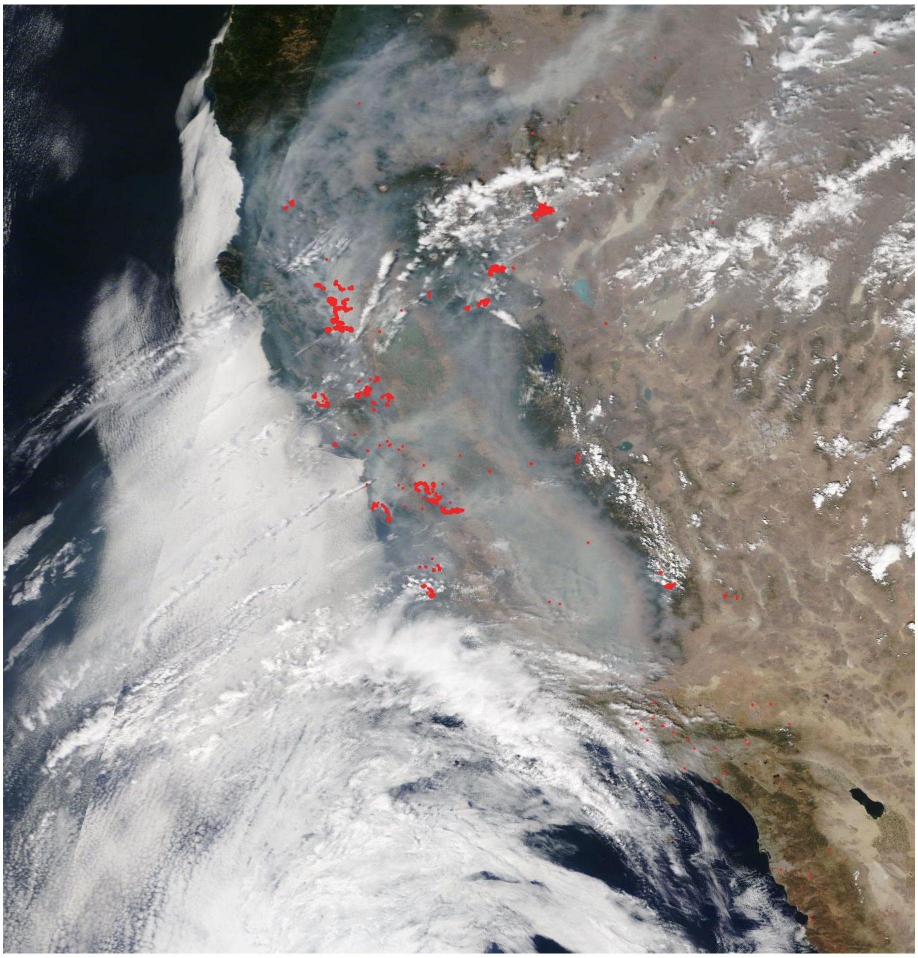


Figure 1.1: NASA's Terra Satellite Shows Smoky Pall Over Most of California. Source:[9]

Methods and Proceidures

2.1 Data-sets and Study Extents

2.1.1 Carbon Monoxide Levels from MOPITT

As we intend on studying the levels of Carbon Monoxide contained at different levels of the atmosphere, we need a public set capable of providing us this data. From a suggestion of our supervisor Pr. Jean Luc Attié, we were advised to look for the data from the MOPITT instrument (Measurement of Pollution in the Troposphere) located on the TERRA satellite from NASA. As it entails, the instrument measures different pollutants levels in the atmosphere along its ground track. It achieves a spatial resolution of $22\ km$ at nadir with a total swath of $640\ km$. The Carbon Monoxide is measured in layers of $5\ km$ in the air column below [8]. This data is accessible through the Earth Data ASDC website [5] in the form of Hierarchical Data Format Release 5 files (.he5), one for each day since $2000/03/03$. However, this data-set poses a storage constraint from its huge volume.
For MOPITT data ([7]), we will only use data from certain columns.

- Latitude : Latitude of the pixel (°)
- Longitude : Longitude of the pixel (°)
- Time : TAI Time of the observation (s)
- Pressure : Pressure Levels (hPa)
- RetrievedCOMixingRatioProfile : CO Mixing Ratio for the layer above each pressure level (ppbv : parts per billion by volume)

2.1.2 Active Fires Locations from MODIS

In conjunction with the CO study, we will analyse the fires and study their impact on the CO levels in the atmosphere. This will be done by retrieving data from the MODIS (Moderate-Resolution Imaging Spectroradiometer) system, launched by NASA. With a resolution of 1 pixel per kilometre, MODIS detects thermal anomaly points in real time. The data is also available on NASA's Earth Science Data System [4] in a CSV (Comma-separated values) format. In different columns, we can get the latitude, longitude, brightness, knowledge or even the moment of the day. These criterion can be vital to judge our data later on. The CSV is structured as shown in the table below ([11]):

Latitude	Longitude	Brightness	Scan	Track	Acq_date	Acq_time	Satellite	Confidence	Version	Bright_t31	frp	Daynight
-23.63776	151.13786	312.95	1.21	1.09	2022-03-24	0013	T	33	6.1NRT	299.7	7.03	D

Figure 2.1: Format of a MODIS CSV

Column	Signification	Example
CSV		
Latitude	Latitude of the measurement	$-23.63776°$
Longitude	Longitude of the measurement	$151.13786°$
Brightness	Measured brightness temperature (in Kelvin) using the MODIS channels 21/22 and channel 31	$312.95\ K$
Scan	The spatial-resolution in the East-West direction of the scan for 1 pixel (in kilometer)	$1.21\ km$
Track	The North-South spatial resolution of the scan for 1 pixel (in kilometer)	$1.09\ km$
Acq_date	Measurement acquisition date	2022-03-24
Acq_time	Time of measurement acquisition (in UTC)	$0013 \longrightarrow 00\ h\ 13\ min$
Satellite	Whether the detection was picked up by the Terra or Aqua satellite	T
Confidence	Quality flag of the individual hotspot/active fire pixel (in %)	$33\ \%$
Version	Refers to the processing collection and source of data	6.1NRT
Bright_t31	Channel 31 brightness temperature (in Kelvins) of the hotspot/active fire pixel	299.7
FRP (Fire Radiative Power)	Depicts the pixel-integrated fire radiative power in MW (MegaWatts). FRP provides information on the measured radiant heat output of detected fires	7.03
Daynight	Specifies whether the measurement was made at night or during the day	D

2.1.3 Study Extents

This work has a focus on the impact on atmospheric carbon monoxide of wildfires that occurred in Summer 2020 in California. In order to get a good picture of the geographical extent of our study, we decided to not constrain ourselves to just California but to get a broader view of the Western coast of the continental United States by defining a square ranging from $-125°$ to $-115°$ in longitude and $30°$ to $45°$ in latitude. This geographical zone can be seen more easily in Fig. 2.2. As for the time periods we will look upon in this study, we primarily analyse the summer 2020, however, to better understand the shear intensity and impact of that year's wildfires, it would be a good idea to compare them to different time periods such as the winter of the same year or a different year. For this purpose, we will also study both the winter of 2020 and the summer of 2015.

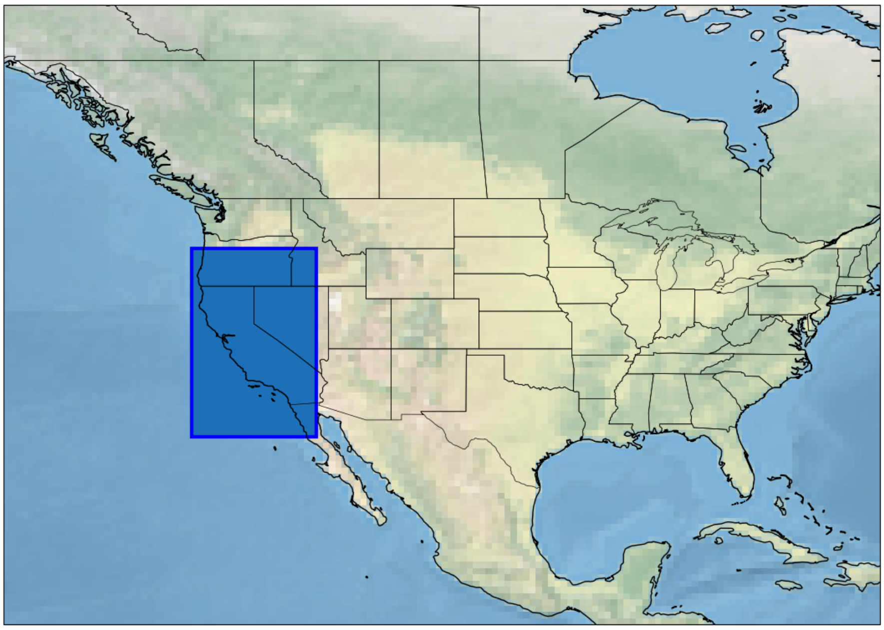


Figure 2.2: Visualization of the geographical study extent

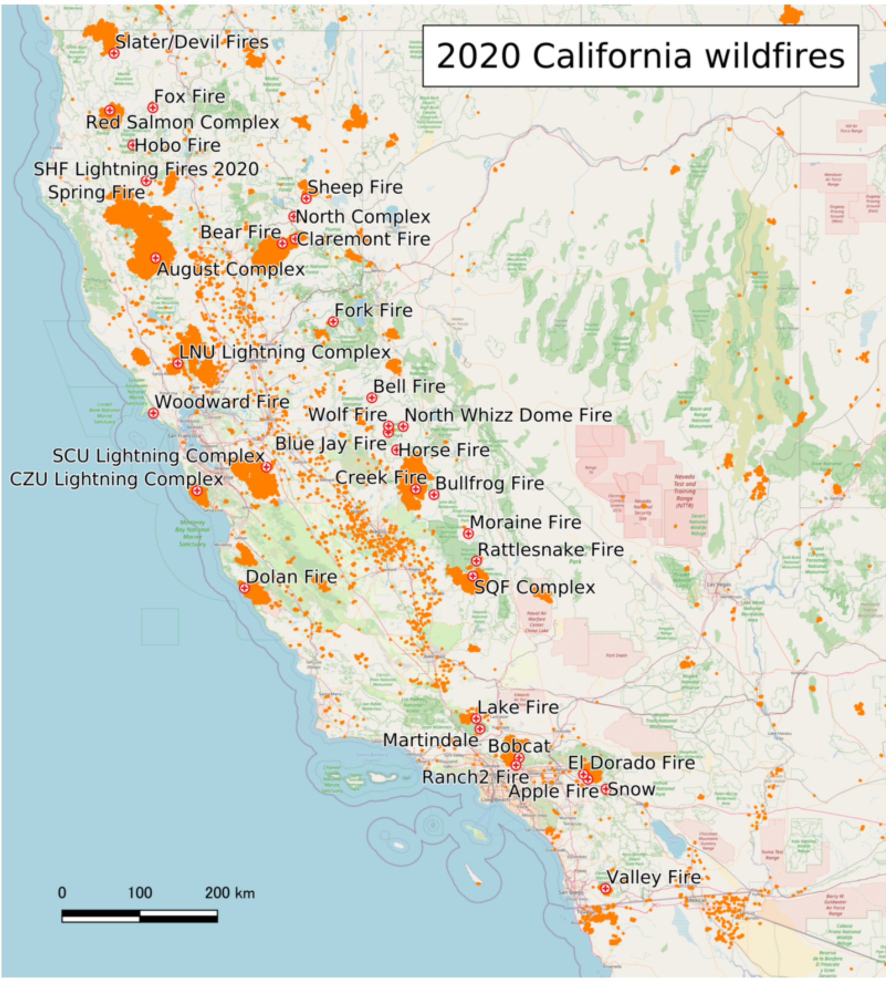


Figure 2.3: Map of 2020 California Wildfires. Source: [1]

2.2 Data Extraction and Processing

2.2.1 Extracting data and averaging for Imaging

We first begin by extracting the carbon monoxide levels from the **he5** files provided from the MOPITT TERRA instrument. We store them in a directory adjacent to our script and traverse each file, extracting the data using the **h5py** Python library and then selecting the columns that are interesting to us, especially the *RetrievedCOMixingRatioProfile*, *Longitude* and *Latitude*. The time of observation does not interest us that much since the data is already organized in a one day per file fashion. The data is selected along a certain pressure level that we have to mention. We will use *pressure level* $= 2$ by default (this corresponds to a pressure around $700\ mbar$). The resulting carbon monoxide mixing ratio is expressed in parts per billion by volume as mentioned previously. The MODIS fire data is in the form of **CSV** files. The data contained in such files can easily be extracted using the **pandas** Python library, especially the **read_csv** function which returns a Dataframe from the mention of the file path. This time, the data is not sorted in a one day per file setup, this mans we have to actively select the data based on the time given in the *Acq_Date*. We also sorted it using the *confidence* factor by only taking data with a 80% confidence of it being a fire. We also could have eliminated the ones spotted during the day. However, this could have been seen as useless.

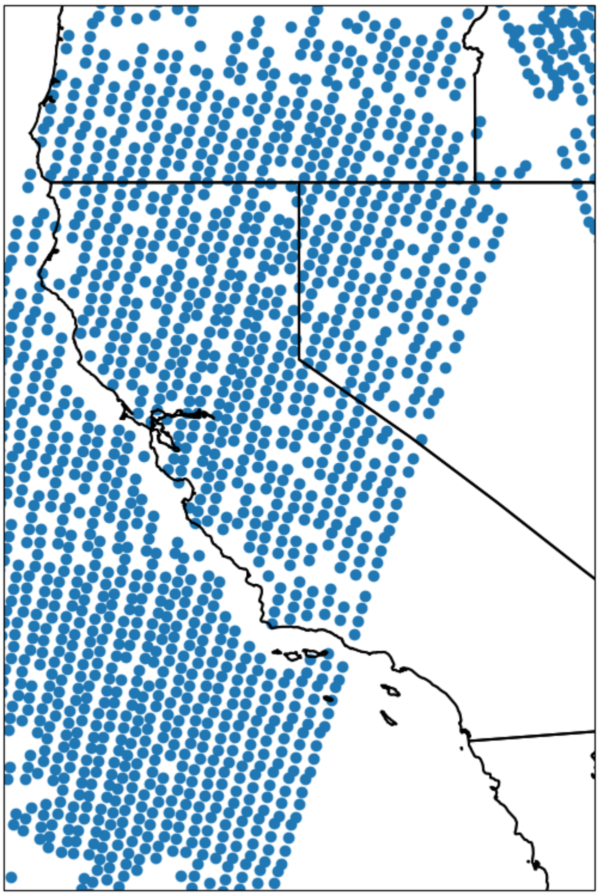


Figure 2.4: Scattered locations view of MOPITT data from 2020-08-01

The main problem with the data from the MODIS and MOPITT is that they consist of values given for certain longitudes and latitudes that were taken as the satellite was passing over these locations. This means that the data is in a scattered form, as can be seen represented in Fig.2.4. However, if we want to be able to study images, we need a method of creating them from these scattered points. A simple method is then to create a longitude/latitude grid in our study extent and then register the points, counting the number in each pixel of this grid and averaging them. In our case, this process is done through a modified version of the **average_grid** Python function originally created by our supervisor Pr. Jean Luc Attié. This function can be found in our source code at the end of this report. This gives us something alike what is shown in Fig.2.5. From these images, we can then get the complete average using the **numpy.nansum** function. We also used the **numpy.corrcoef** function in order to study the correlation matrices resulting from the confrontation of the wildfires and carbon monoxide data.

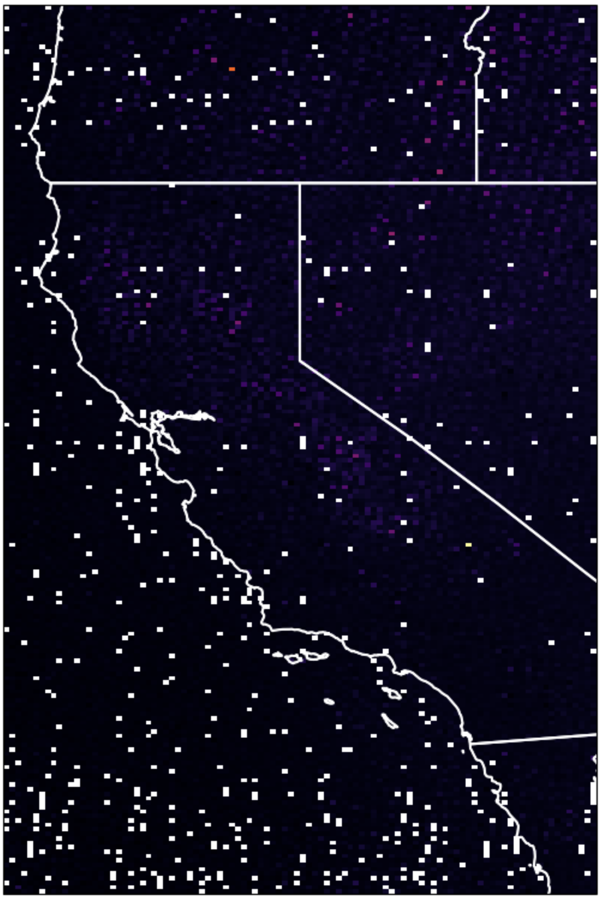


Figure 2.5: Averaged view of MOPITT data from 2020-08-01 to 2020-10-31

2.2.2 Plotting with Cartopy

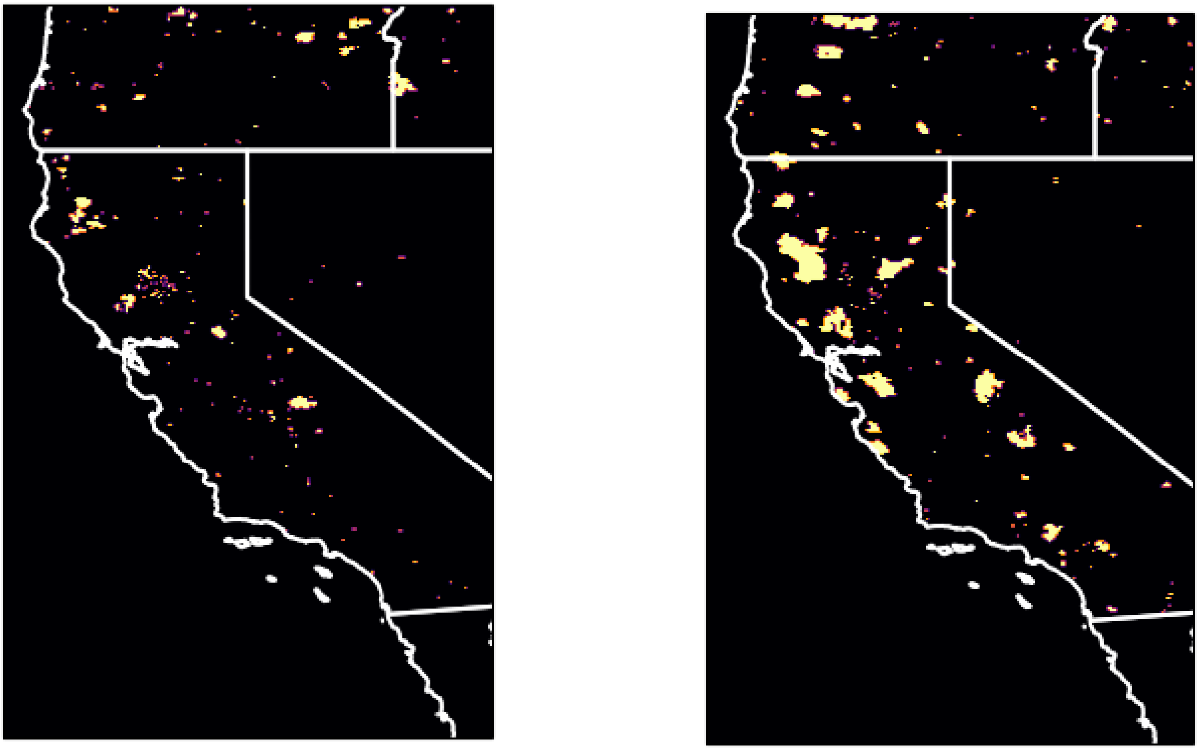
Cartopy is a Python library set to expand the abilities of the **Matplotlib** Python library by allowing users to plot images, scattered data, lines and other types of data on geographical renditions. This is the main tool we have used to plot the different geographical views shown in the next section focused on our results. These have been set to comply with the geographical extent we have set up previously as shown in Fig. 2.2 and have used the *PlateCarree* projection. It allow us to keep the principles of the *Mercator* projection without having the huge differences in surfaces areas when comparing the lands in the Polar circles and those at the Equator. **Cartopy** also gives us the ability to show the coastlines, national and state border and allows for the addition of rivers, lakes and stock images of satellite imagery.

Results

3.1 Study of the amount of wildfires

3.1.1 Initial Wildfires surface area Validation

We first decided to display the wildfires surface area by counting all data points on the entire time extent of our data as a way of showing the entire burnt areas. This results in the images shown in the figures right below. They show data from Summer 2020 and Summer 2015. A huge difference in the amount of afflicted areas can already be pointed out but another important aspect of our study can be concluded from these images. Indeed, if we compare these with what is shown in Fig 2.2, the same fire distribution can be seen with a significant resemblance. Since the Wikipedia source image we used was also done using the MODIS data, this grants us a way of validating the averaging/extraction process we detailed in the previous section. It's easy to spot some very large perturbations in the 2020 period, notably the SCU Lightning Complex, the August Complex and LNU Lightning Complex wildfires. The difference between the amounts of burnt areas is shocking. This shows the sheer impact the 2020 California wildfires had.



(a) Wildfires Map for Summer 2015

(b) Wildfires Map for Summer 2020

3.1.2 Comparison of the amounts of wildfires in time-series

The amount of lit up pixels in our images were plotted in Fig.3.2 and Fig.3.3 for the Summer 2020 and Summer 2015 time periods. The X axis show the time in days since the first day given by our data-sets.

For the 2015 time period, a huge increase can be seen in August with a drop in September, leaving practically no fires in October. As for Summer 2020 however, the biggest amounts were registered in late August and September. A huge difference in the amount of fires can be registered by comparing these time series. A factor of 6 can be observed between the maximums of both graphs (shown explicitly with the use of the red dashed lines). The averages are also significantly different, giving us an approximated average of 40 pixels for 2015 and of 100 for 2020.

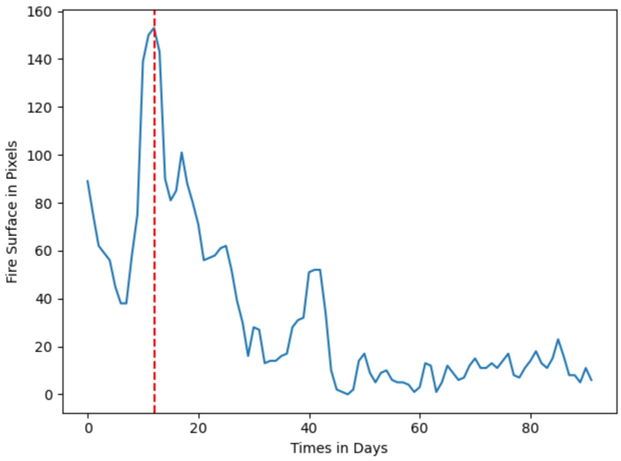


Figure 3.2: Evolution of fires from August to October 2015

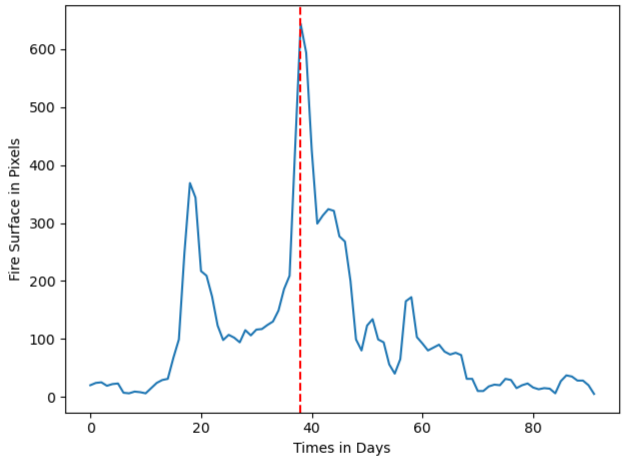


Figure 3.3: Evolution of fires from August to October 2020

The high points shown in Fig.3.3 derives from the so-called "August Complex fire" and "Creek Fire". These specific fires started on September 4th, 2020 until December 24th, 2020 and burned over 380 000 Acres. It took place mainly to the northern California and West of San Francisco, as can be seen in the figure 2.3.

3.2 Study and Comparison of the Carbon Monoxide Levels

The average Carbon Monoxide Levels were averaged over each day. The resulting graphs are shown in Fig.3.4 and Fig. 3.5. Firstly, a similar abundance of Carbon Monoxide can be observed in both summer time periods. Indeed, the usual summer level seems to be set around $80\ ppbv$. However, a significant increase in the general levels of the "spikes" can be seen for the Summer 2020 period. It seems to variate around $110\ ppbv$ when the Summer 2015 period shows a spike average of $90\ ppbv$. However, this does not show a link between the intensity of wildfires and carbon monoxide levels. This statement could be asserted from a correlation study of the data.

We also wanted to plot the Carbon Monoxide levels shown in Winter 2020 to see the possible differences between Summer and Winter. The resulting graph is shown in Fig.3.6. An average level of $95\ ppbv$ can be observed during this time period, far higher than the $80\ ppbv$ shown during the Summer. This could be explained in part by the majority of California citizens relying on natural gas or fuel to heat their homes[6], resulting in a higher level of released pollutants in the atmosphere. Also, a surprising "peak" can be seen on the 27th day of this time series (the 28th of January 2020). We found a few plausible explanations such as the tribute given to Kobe Bryant near Los Angeles after his death. A sudden influx of fans was reported by the media at that time [10].

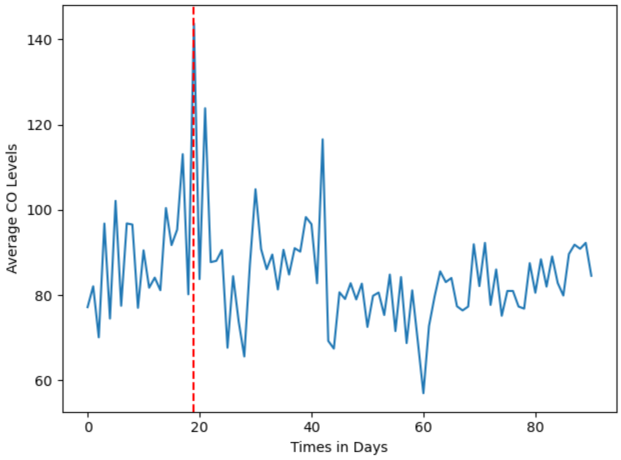


Figure 3.4: CO Levels in Summer 2015

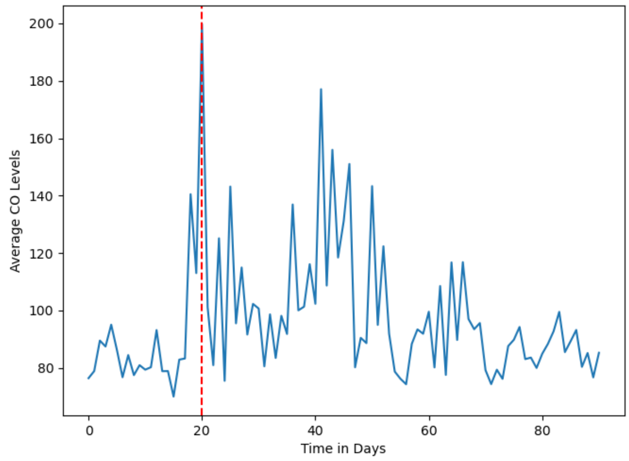


Figure 3.5: CO Levels in Summer 2020

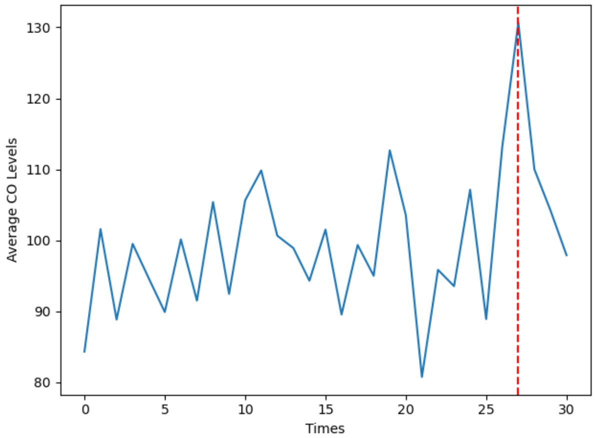


Figure 3.6: CO Levels in Winter 2020

3.3 Correlation between Wildfires and Atmospheric CO Levels

By studying the correlation matrix between our carbon monoxide levels and our wildfires amounts, a way of concluding on the link between them could be asserted. Firstly, we wanted to plot the previously shown CO levels and wildfires pixel counts on the same graphs to get a better view of their plausible link. This resulted in the graphs shown in Fig.3.7 and Fig.3.8. We made sure of normalizing them using their maximums to fit them in the same graphs. Each time, we can observe a peak of CO abundance following a peak in wildfires surface area. It is especially blatant in the Summer 2020 case.
We also decided on getting access to the correlation matrices of both summer time series:

$$\begin{bmatrix} 1 & 0.41687387 \\ 0.41687387 & 1 \end{bmatrix}_{2015} \qquad \begin{bmatrix} 1 & 0.70600171 \\ 0.70600171 & 1 \end{bmatrix}_{2020}$$

For the summer of 2020, we get a correlation of around 70% against a feeble 40% for Summer 2015. This gives us more leverage to claim the CO levels observed in the atmosphere were due to the amount of wildfires for the 2020 Summer period.

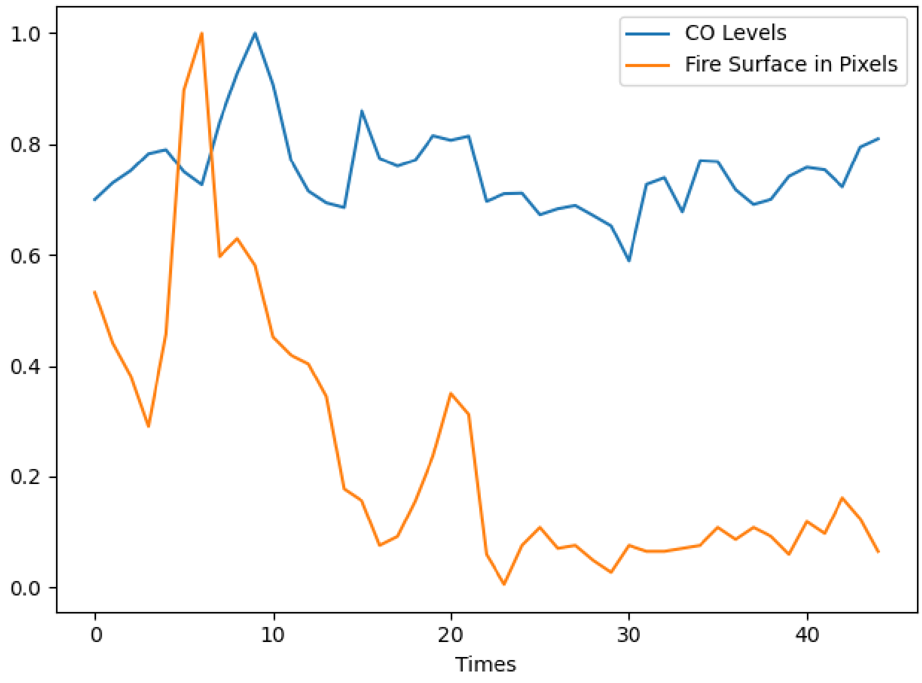


Figure 3.7: Comparison between the fire surface area and the CO level for the summer 2015 period.

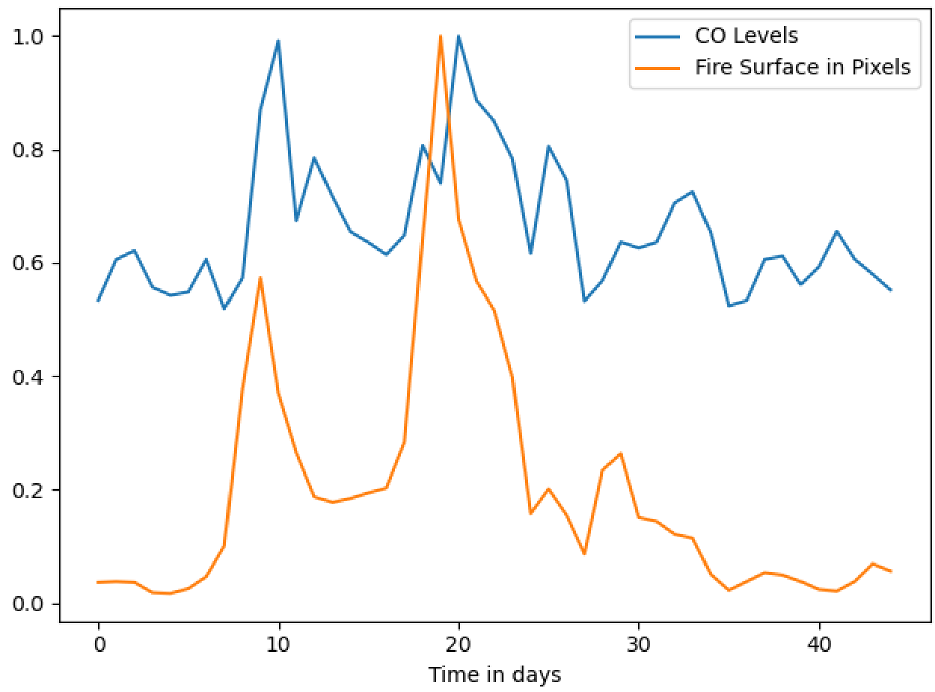


Figure 3.8: Comparison between the fire surface area and the CO level for the summer 2020 period.

Conclusion

This homework project allowed us to have a general overview of Python's many possibilities in Earth observation. It has been an efficient and strong open source tool, perfect for users with modest skills that want access to powerful and generally useful digital image processing techniques.

The 2020 Western United States wildfire season was chosen to grasp the growing impact of these disasters, given the increase in droughts caused by climate change and the recent lack of controlled fires to remove dry biomass[1]. Our findings proved the expected correlation between the wildfires' surface area and carbon monoxide abundance in the atmosphere. In the interest of our study, the overall decrease in the yearly CO emissions of the entire US territory [2] evolves opposite to the tendency of wildfires. We can therefore conclude that these natural disasters are responsible for an increasingly important portion in the general national emissions of carbon monoxide, accentuating the need for better forest management in the presence of an increasing climate change risk.

Most of this project's difficulties came from finding the data, understanding it, extracting and use as well as creating a reliable code which could be used swiftly and efficiently. The increasing number of other reports we were subjected to produce this year greatly impacted the reduced amount of time we could spend on this project. Given more time, we could have further developed this subject analysis, by studying the layout of vegetation in California and its link to carbon monoxide, but also the study of the impact of wildfires on Nitrous Oxide in the atmosphere.

References

[1] *2020 California wildfires - Wikipedia*. URL: https://en.wikipedia.org/wiki/2020_California_wildfires.

[2] *Annual carbon monoxide emissions in the United States from 1970 to 2020*. URL: https://www.statista.com/statistics/501274/volume-of-carbon-monoxide-emissions-us/#:~:text=Carbon%5C%20monoxide%5C%20(CO)%5C%20emissions%5C%20in,decline%5C%20in%5C%20the%5C%20following%5C%20years..

[3] *California-Wildfires — Github Repository*. URL: https://github.com/EnguerranVidal/California-Wildfires.

[4] *Earth Science Data System*. URL: https://earthdata.nasa.gov/learn/toolkits/wildfires.

[5] *EarthData ASDC*. URL: https://asdc.larc.nasa.gov/data/MOPITT/.

[6] *Heating Systems used in Southern California*. URL: https://dynamicairservices.com/blog/best-heating-system-southern-california/.

[7] *MOPITT (Measurements of Pollution in the Troposphere) Version 8 Product User's Guide*. URL: https://www2.acom.ucar.edu/sites/default/files/mopitt/v8_users_guide_201812.pdf.

[8] *MOPITT — Terra*. URL: https://terra.nasa.gov/about/terra-instruments/mopitt.

[9] *NASA's Terra Satellite Shows Smoky Pall Over Most of California*. URL: https://www.nasa.gov/feature/goddard/2020/nasa-s-terra-satellite-shows-smoky-pall-over-most-of-california.

[10] *Newspaper front pages a day after Kobe Bryant's death*. URL: https://www.tampabay.com/sports/2020/01/27/newspaper-front-pages-a-day-after-kobe-bryants-death/.

[11] *Satellite (MODIS) Thermal Hotspots and Dire Activity*. URL: https://www.arcgis.com/home/item.html?id=b8f4033069f141729ffb298b7418b653.

Codes

All codes and CSV data will be on our Github repository [3]. The data directories and CSV filenames can be changed to fit the needs of the user.

```python

import numpy as np
import matplotlib.pyplot as plt
import h5py
import cartopy.crs as ccrs
import cartopy.feature as cfeature
import os

import datetime as dt
import pandas as pd
import imageio


def average_grid(val_data, val_long, val_lat, long, lat, flipped=True, cropped=True):
    count = np.zeros((lat.shape[0] - 1, long.shape[0] - 1))
    average = np.zeros((lat.shape[0] - 1, long.shape[0] - 1))
    n = len(val_data)
    for i in range(n):
        if val_data[i] != np.nan and val_data[i] > 0:
            if long[0] < val_long[i] < long[-1] and lat[0] < val_lat[i] < lat[-1]:
                index_long = np.digitize(val_long[i], long) - 1
                index_lat = np.digitize(val_lat[i], lat) - 1
                average[index_lat, index_long] += val_data[i]
                count[index_lat, index_long] += 1
    valid = count != 0
    average[valid] = average[valid] / count[valid]
    if cropped:
        cropping = np.where(count == 0)
        average[cropping] = np.nan
    if flipped:
        return np.flip(average, axis=0), np.flip(count, axis=0)
    else:
        return average, count


def h5_MOPITT_loader(dir_path, extent, size, averaging=1, n_pressure=2):
    averages, counts = [], []
    long = np.linspace(extent[0], extent[1], size[0])
    lat = np.linspace(extent[2], extent[3], size[1])
    DATAFIELD_NAME = '/HDFEOS/SWATHS/MOP02/Data Fields/RetrievedCOMixingRatioProfile'
    GEO_DATA = '/HDFEOS/SWATHS/MOP02/Geolocation Fields'
    files = [f for f in os.listdir(dir_path) if os.path.isfile(os.path.join(dir_path, f))]
    n_snaps = int(len(files) / averaging)
    for i in range(n_snaps):
        values, longitudes, latitudes = [], [], []
        for j in range(averaging):
            index = i * averaging + j
            path = os.path.join(dir_path, files[index])
            with h5py.File(path, mode='r') as file:
                # Extract Datasets
                data_var = file[DATAFIELD_NAME]
                data_lat = file[GEO_DATA + '/Latitude']
                data_lon = file[GEO_DATA + '/Longitude']
                # Read Values
                val_lat = data_lat[:]
                val_lon = data_lon[:]
                val_data = data_var[:, n_pressure, 0]
                longitudes += val_lon.tolist()
                latitudes += val_lat.tolist()
                values += val_data.tolist()
```

```python
        average, count = average_grid(values, longitudes, latitudes, long, lat)
        averages.append(average)
        counts.append(count)
    return averages, counts


def simple_plot_map(matrix, extent, borderlines="white"):
    ax = plt.axes(projection=ccrs.PlateCarree())
    ax.set_extent(extent)
    ax.coastlines(color=borderlines)
    ax.add_feature(cfeature.STATES, zorder=1, linewidth=1.5, edgecolor=borderlines)
    ax.imshow(matrix, transform=ccrs.PlateCarree(), extent=extent, cmap='inferno')
    plt.show()


def csv_MODIS_loader(file_path, extent, size, averaging=1, beginning="2020-08-01"):
    long = np.linspace(extent[0], extent[1], size[0])
    lat = np.linspace(extent[2], extent[3], size[1])
    date_format = "%Y-%m-%d"
    df = pd.read_csv(file_path)
    df["acq_date"] = pd.to_datetime(df["acq_date"], format=date_format)
    df["acq_date"] = pd.to_datetime(df["acq_date"], format=date_format)
    n_snaps = int(92 / averaging)
    beginning = dt.datetime.strptime(beginning, date_format)
    averages, counts, times = [], [], []
    for i in range(n_snaps):
        start_time = beginning + dt.timedelta(days=averaging * i)
        end_time = beginning + dt.timedelta(days=averaging * (i + 1))
        if averaging == 1:
            times.append(start_time.strftime("%Y-%m-%d"))
        else:
            times.append(start_time.strftime("%Y-%m-%d") + " to " + end_time.strftime("%Y-%m-%d"))
        mask = (df["acq_date"] >= start_time) & (df["acq_date"] <= end_time) & (df["confidence"] >
    80)
        data = df.loc[mask]
        latitude = np.array(data["latitude"])
        longitude = np.array(data["longitude"])
        brightness = np.array(data["brightness"])
        average, count = average_grid(brightness, longitude, latitude, long, lat, cropped=False)
        averages.append(average)
        counts.append(count)
    return averages, counts, times


def create_fires_gif_map():
    average = 1
    MODIS_filename = "fire_archive_M-C61_245017.csv"
    extent = [-125, -115, 30, 45]
    size = [201, 401]
    averages, counts, times = csv_MODIS_loader(MODIS_filename, extent, size, averaging=average)
    images_files = []
    print("Start of Loop - Creating Images")
    n = len(counts)
    for i in range(n):
        ax = plt.axes(projection=ccrs.PlateCarree())
        ax.set_title(times[i])
        ax.set_extent(extent)
        ax.coastlines(color="white")
        ax.add_feature(cfeature.STATES, zorder=1, linewidth=1.5, edgecolor="white")
        ax.imshow(averages[i], transform=ccrs.PlateCarree(), extent=extent, cmap='inferno')
        filename = f'Day_{i}.png'
        images_files.append(filename)
        plt.savefig(filename, dpi=150)
        plt.close()
        print(f'{i * 100 / n:.2f} %')
    print("End of Loop - Creating GIF")
    # Creating the GIF
    with imageio.get_writer('Fire.gif', mode='I', fps=12) as writer:
        for filename in images_files:
            image = imageio.imread(filename)
            writer.append_data(image)
    writer.close()
    # Delete Old Images
```

```
    for filename in set(images_files):
        os.remove(filename)
    print("Finished Creating GIF")


def plot_fire_levels():
    average = 1
    MODIS_filename = "fire_archive_M-C61_245017.csv"
    extent = [-125, -115, 30, 45]
    size = [201, 401]
    averages, counts, times = csv_MODIS_loader(MODIS_filename, extent, size, averaging=average)
    counts_list = []
    n = len(counts)
    for i in range(n):
        ones = counts[i] > 0
        counts[i][ones] = 1
        counts_list.append(np.nansum(counts[i]))
    times = np.arange(n)
    fig, ax = plt.subplots(1, 1)
    ax.plot(times, counts_list)
    ax.set_xlabel("Times")
    ax.set_ylabel("Fire Surface in Pixels")
    ax.axvline(x=counts_list.index(max(counts_list)), color='red', linestyle='--')
    plt.tight_layout()
    plt.show()


def extent_map():
    import matplotlib.patches as mpatches
    extent = [-140, -70, 15, 60]
    ax = plt.axes(projection=ccrs.PlateCarree())
    ax.set_extent(extent)
    ax.coastlines(color="black")
    ax.stock_img()
    ax.add_patch(mpatches.Rectangle(xy=(-125, 30), width=10, height=15, lw=3,
                                    edgecolor='blue', transform=ccrs.PlateCarree()))
    ax.add_patch(mpatches.Rectangle(xy=(-125, 30), width=10, height=15, facecolor="blue", alpha
    =0.05,
                                    transform=ccrs.PlateCarree()))
    ax.add_feature(cfeature.STATES, zorder=1, linewidth=0.5, edgecolor="black")
    plt.show()


def plot_CO_levels():
    average = 1
    MOPPIT_data_directory = os.path.join(os.path.dirname(os.path.abspath(__file__)), "data")
    extent = [-125, -115, 30, 45]
    size = [201, 401]
    P = 2
    averages, counts = h5_MOPITT_loader(MOPPIT_data_directory, extent, size, averaging=average,
    n_pressure=P)
    counts_list = []
    n = len(counts)
    for i in range(n):
        counts_list.append(np.nansum(averages[i]) / np.nansum(counts[i]))
    times = np.arange(n)
    fig, ax = plt.subplots(1, 1)
    ax.plot(times, counts_list)
    ax.set_xlabel("Time in Days")
    ax.set_ylabel("Average CO Levels")
    ax.axvline(x=counts_list.index(max(counts_list)), color='red', linestyle='--')
    plt.tight_layout()
    plt.show()


def all_fires():
    average = 92
    MODIS_filename = "fire_archive_M-C61_245017.csv"
    extent = [-125, -115, 30, 45]
    size = [201, 401]
    averages_fires, counts_fires, times_fires = csv_MODIS_loader(MODIS_filename, extent, size,
    averaging=average)
    counts = counts_fires[0]
```

```
    ones = counts > 1
    counts[ones] = 1
    simple_plot_map(counts, extent, borderlines="white")


def confront_CO_fire():
    average = 2
    MODIS_filename = "fire_archive_M-C61_245017.csv"
    MOPPIT_data_directory = os.path.join(os.path.dirname(os.path.abspath(__file__)), "data")
    extent = [-125, -115, 30, 45]
    size = [201, 401]
    P = 2
    averages_CO, counts_CO = h5_MOPITT_loader(MOPPIT_data_directory, extent, size, averaging=average
, n_pressure=P)
    averages_fires, counts_fires, times_fires = csv_MODIS_loader(MODIS_filename, extent, size,
averaging=average)
    CO_list = []
    Fires_list = []
    n = len(counts_CO)
    for i in range(n):
        ones = counts_fires[i] > 0
        counts_fires[i][ones] = 1
        CO_list.append(np.nansum(averages_CO[i]) / np.nansum(counts_CO[i]))
        Fires_list.append(np.nansum(counts_fires[i]))
    times = np.arange(n)
    fig, ax = plt.subplots(1, 1)
    ax.plot(times, np.array(CO_list) / max(CO_list), label="CO Levels")
    ax.plot(times, np.array(Fires_list) / max(Fires_list), label="Fire Surface in Pixels")
    ax.set_xlabel("Time in days")
    # ax.axvline(x=Fires_list.index(max(Fires_list)), color='red', linestyle='--')
    plt.tight_layout()
    plt.legend()
    plt.show()
    # Add Correlation Coefficients between CO and fire data
    corr = np.corrcoef(np.array(CO_list) / max(CO_list), np.array(Fires_list) / max(Fires_list))
    print(corr)


def plot_weeks():
    average = 7
    MODIS_filename = "fire_archive_M-C61_245017.csv"
    MOPPIT_data_directory = os.path.join(os.path.dirname(os.path.abspath(__file__)), "data")
    extent = [-125, -115, 30, 45]
    size = [201, 401]
    P = 2
    averages_CO, counts_CO = h5_MOPITT_loader(MOPPIT_data_directory, extent, size, averaging=average
, n_pressure=P)
    averages_fires, counts_fires, times_fires = csv_MODIS_loader(MODIS_filename, extent, size,
averaging=average)
    CO_list = []
    Fires_list = []
    n = len(counts_CO)
    for i in range(n):
        CO_list.append(averages_CO[i])
        Fires_list.append(averages_fires[i])
    # Fires Weekly Plots
    plt.figure(figsize=(8, 10))
    axes = []
    for i in range(12):
        axes.append(plt.subplot(4, 3, i + 1, projection=ccrs.PlateCarree()))
        axes[i].set_extent(extent)
        axes[i].coastlines(color="white")
        axes[i].add_feature(cfeature.STATES, zorder=1, linewidth=0.25, edgecolor="white")
        axes[i].imshow(Fires_list[i], transform=ccrs.PlateCarree(), extent=extent, cmap='inferno')
    plt.show()
    # CO Weekly Plots
    plt.figure(figsize=(8, 10))
    axes = []
    for i in range(12):
        axes.append(plt.subplot(4, 3, i + 1, projection=ccrs.PlateCarree()))
        axes[i].set_extent(extent)
        axes[i].coastlines(color="black")
        axes[i].add_feature(cfeature.STATES, zorder=1, linewidth=0.25, edgecolor="black")
```

```
        axes[i].imshow(CO_list[i], transform=ccrs.PlateCarree(), extent=extent, cmap='inferno')
    plt.show()


def scattered_data():
    averaging = 1
    n_pressure = 2
    MOPPIT_data_directory = os.path.join(os.path.dirname(os.path.abspath(__file__)), "data")
    DATAFIELD_NAME = '/HDFEOS/SWATHS/MOP02/Data Fields/RetrievedCOMixingRatioProfile'
    GEO_DATA = '/HDFEOS/SWATHS/MOP02/Geolocation Fields'
    files = [f for f in os.listdir(MOPPIT_data_directory) if os.path.isfile(os.path.join(
    MOPPIT_data_directory, f))]
    n_snaps = int(len(files) / averaging)
    values, longitudes, latitudes = [], [], []
    for i in range(n_snaps):
        values, longitudes, latitudes = [], [], []
        for j in range(averaging):
            index = i * averaging + j
            path = os.path.join(MOPPIT_data_directory, files[index])
            with h5py.File(path, mode='r') as file:
                # Extract Datasets
                data_var = file[DATAFIELD_NAME]
                data_lat = file[GEO_DATA + '/Latitude']
                data_lon = file[GEO_DATA + '/Longitude']
                # Read Values
                val_lat = data_lat[:]
                val_lon = data_lon[:]
                val_data = data_var[:, n_pressure, 0]
                longitudes += val_lon.tolist()
                latitudes += val_lat.tolist()
                values += val_data.tolist()
        borderlines = "black"
        ax = plt.axes(projection=ccrs.PlateCarree())
        ax.set_extent(extent)
        ax.coastlines(color=borderlines)
        ax.add_feature(cfeature.STATES, zorder=1, linewidth=1.5, edgecolor=borderlines)
        plt.scatter(x=longitudes, y=latitudes, transform=ccrs.PlateCarree())
        plt.show()
```