Video processing pipeline to concat and accelerate flood videos from Google Cloud Storage bucket

Choose base directory

In [1]:

```
cd ../
```

```
C:\Users\luisr\Desktop\Repositories\Data Science Projects\Hackaton COR IV - Centro de Operações do RJ\INCUBAÇÃO\Cameras
```

Define utility functions

In [13]:

```
from time import time

# Simples class to report execution time

class Timer:
    def __init__(self):
        self.start = time()
    def end(self, decimals=4):
        end = time() - self.start
        print('\n* TIME TO EXECUTE:', round(end, decimals), 's')

# Get blob count, bytes and names from Google Cloud Storage bucket

def gcs_folder_info(folder, ext, bucket_name, print_each=1000):
    prefix = folder
    delimiter = None
    names = []
    timer = Timer()
    blobs = gcs.list_blobs(prefix, delimiter, bucket_name)
    for i, blob in enumerate(blobs):
        if blob.name.endswith(ext):
            names.append([blob.name, blob.size])
        if (i + 1) % print_each == 0: print(f'\n- Blobs Searched: {i + 1}'); co(True)
    names = pd.DataFrame(names, columns=['blob_name', 'bytes']) # build blobs dataframe
    print(f'\n- Blobs Searched: {i + 1}')
    print(f'\n  · Blobs (Matched): {len(names)}')
    print(f'\n  · Giga Bytes (Matched): {round(names["bytes"].sum() / 1e9, 3)} GB')
    timer.end() # prints time to execute
    return names
```

Pipeline methods set up

Import Google Cloud Storage wrapper module and set storage instance

In [3]:

```
from modules.googlecloudstorage import GCS

sa_json = '../../../../Apps/APIs/octa-api/credentials/octacity-iduff.json'
user_project = None
default_bucket_name = 'flood-video-collection'

gcs = GCS(sa_json, user_project, default_bucket_name)
```

Video writer class instance

In [4]:

```
from modules.video import VideoWriter

writer = VideoWriter(fps=3, shape=(854, 480), codec='mp4v')
```

```
c:\Users\luisr\anaconda3\lib\site-packages\pandas\core\computation\expressions.py:20: UserWarning:
Pandas requires version '2.7.3' or newer of 'numexpr' (version '2.7.1' currently installed).
  from pandas.core.computation.check import NUMEXPR_INSTALLED
```

Accelerated video writer class instance

In [5]:

```
from modules.video import VideoWriter

writer_speed = VideoWriter(fps=24, shape=(854, 480), codec='mp4v')
```

Video writer class funcitonality

1. Add running clock to video files
2. Concatenate video files from nested folders
3. Accelerate video files from nested folders

Count blobs with .mp4 extension and total file bytes of download

Import python modules

In [7]:

```
import pandas as pd
from IPython.display import clear_output as co
```

Pipeline Execution

Step 0 · Pipeline parameters set up

Download from Google Cloud Storage bucket

In [46]:

```
'''
Note: Forward trailing slashses, i.e. `/`, at the end of `prefix` limits
results to folders matching exactly to `prefix`. Otherwise, matches any folder or blob
that contains `prefix`.
'''
bucket_name = 'flood-video-collection'
prefix = 'polygons/flood/'
delimiter = None
folder = 'Dados/flood-video-collection' # bucket collection destination folder
report_freq = 5
overwrite_download = False
```

Annotate videos timestamps

In [47]:

```
folder = 'Dados/flood-video-collection'  # local collection source folder
to_folder = 'Dados/flood-video-collection-stamped' # `time-stamped` local collection source folder
ext = '.mp4' # video file format to search for in nested folders
overwrite_annot = False
```

Concatenate and accelerate videos from folders

In [48]:

```
base_folder = 'Dados/flood-video-collection-stamped' # `time-stamped` local collection source folder
to_base_folder = 'Dados/flood-video-collection-date' # concatenated local collection destination folder
overwrite_concat = False
```

General purposes parameters

In [49]:

```
ext = '.mp4'
```

Step 0.1 · List and count blobs and download bytes · *Preparation Step*

In [42]:

```
folder_info = gcs_folder_info(prefix, ext, bucket_name, print_each=1000)
```

```
- Blobs Searched: 1181

  · Blobs (Matched): 1181

  · Giga Bytes (Matched): 0.783 GB

* TIME TO EXECUTE: 1.0821 s
```

Step 1 · Download blobs in Cloud Storage bucket to folder

Step 1.1 · Download blobs in `bucket_name` matching `prefix` to local `folder`

In [43]:

```
timer = Timer()

gcs.download_to_folder(folder, prefix, delimiter, bucket_name, report_freq, overwrite_download)

timer.end()
```

```
PREFIX: polygons/flood/ · RUNNING: 13.5 min · RATE: 0.6854 s / file · FINISH-ESTIMATE: 0.0 min · PR
OGRESS: 1180/1181 · DOWNLOADS: 995/1181

* TIME TO EXECUTE: 810.7053 s
```

Step 2 · Annotate videos with dinamic timestamps

Add clock timestamp to nested video files in `folder`

In [45]:

```
timer = Timer()

writer.annot_folder_nested(folder, to_folder, ext, overwrite_annot)

timer.end()
```

```
VIDEO TIMESTAMP ANNOTATION · DONE: 996/996 · SUCCESS: 996/996

* TIME TO EXECUTE: 264.6119 s
```

Step 3 · Concatenate and accelerate videos from folders

Concatenate videos by date from nested folders in `base_folder`

In [50]:

```
timer = Timer()

writer_speed.concatenate_videos_from_nested_folders_by_date(
    base_folder, to_base_folder, ext, overwrite_concat
)

timer.end()
```

```
 CONCAT VIDEOS FROM FOLDER BY DATE · DONE: 57/57 · FOLDER: polygons/flood/8/267

* TIME TO EXECUTE: 220.7912 s
```