

A satellite map of a conflict area, likely in the Middle East, showing a river and surrounding terrain. Red and cyan markers are overlaid on the map, indicating specific locations of interest. The title text is centered over the map.

Monitoring Conflict Areas with SatImaging Time Series

Final Presentation — Project Results

01.02.2019

Christopher, Yousef, Jonathan, Albert, Raoul, Jan, Norwin

Outline

- Motivation
- Project goals
- `sentinel_fire`
 - architecture
 - processing steps
- Results
- Challenges & future work

Motivation



[3] Burning village in Myanmar



[2] Burning village in the Jebel Marra region in Darfur



Project Goals

1

Creation of a satellite image time series processing pipeline

parameterized - parallelized - reproducible - cached

2

Evaluation & development of detection methodologies

detecting burned areas with Sentinel 2 imagery

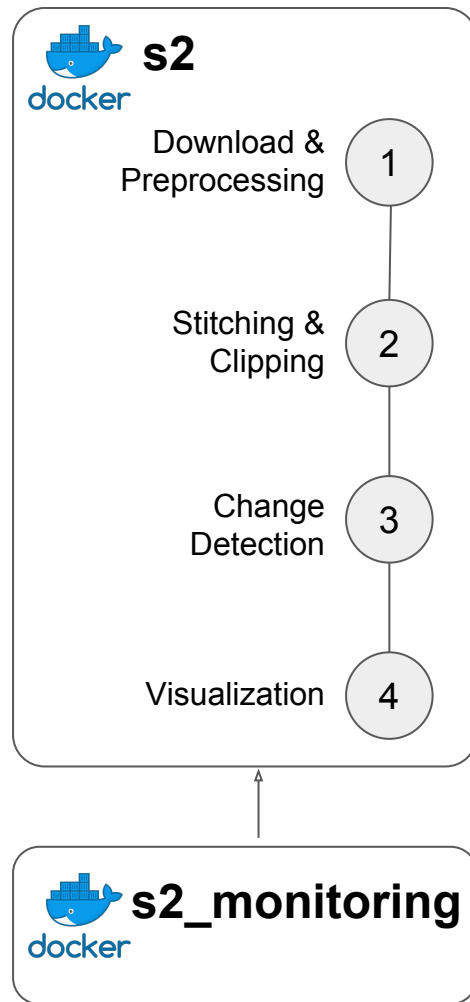
3

Visualization of Results

visual result exploration — understandable, ease of use

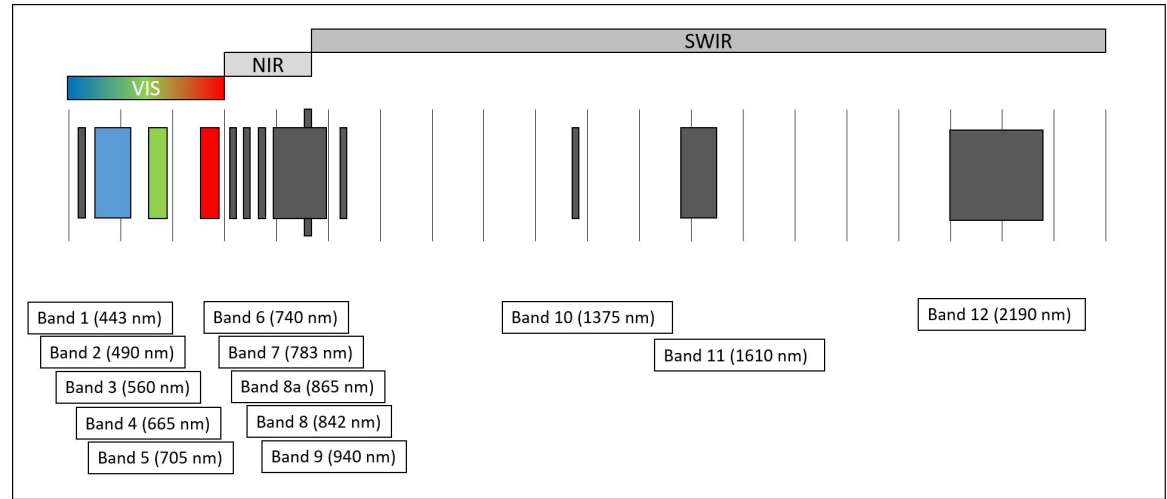
Architecture

- pipeline
 - command line interface
 - result caching
 - parallelism through bash pipes
 - language agnostic for each processing step
- dockerized deployment
 - reproducibility
 - simplicity
- dockerized monitoring
 - cronjob processing latest imagery



Data Source

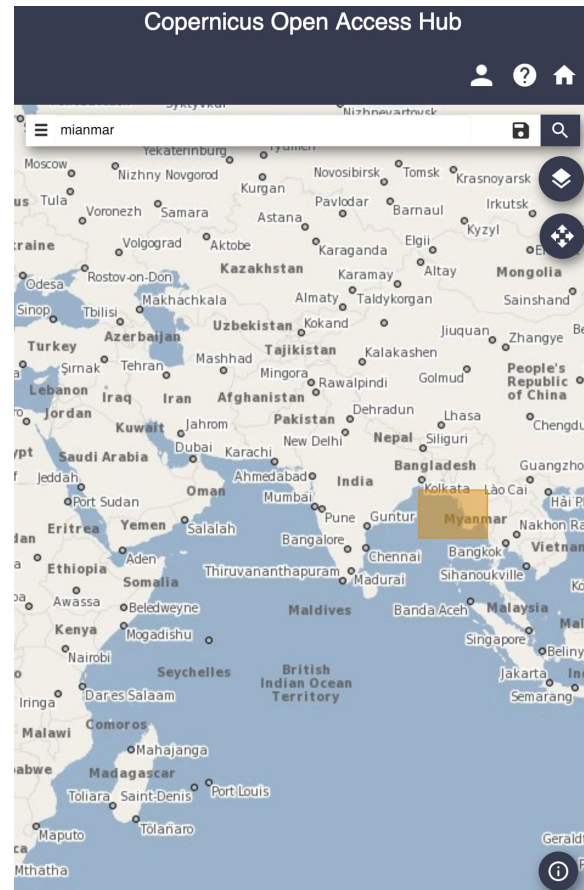
- Sentinel 2 imagery
- high resolution
- variety of bands in visible & infrared
 - comparable to Landsat 8



→ reusability of established methods for burned area detection

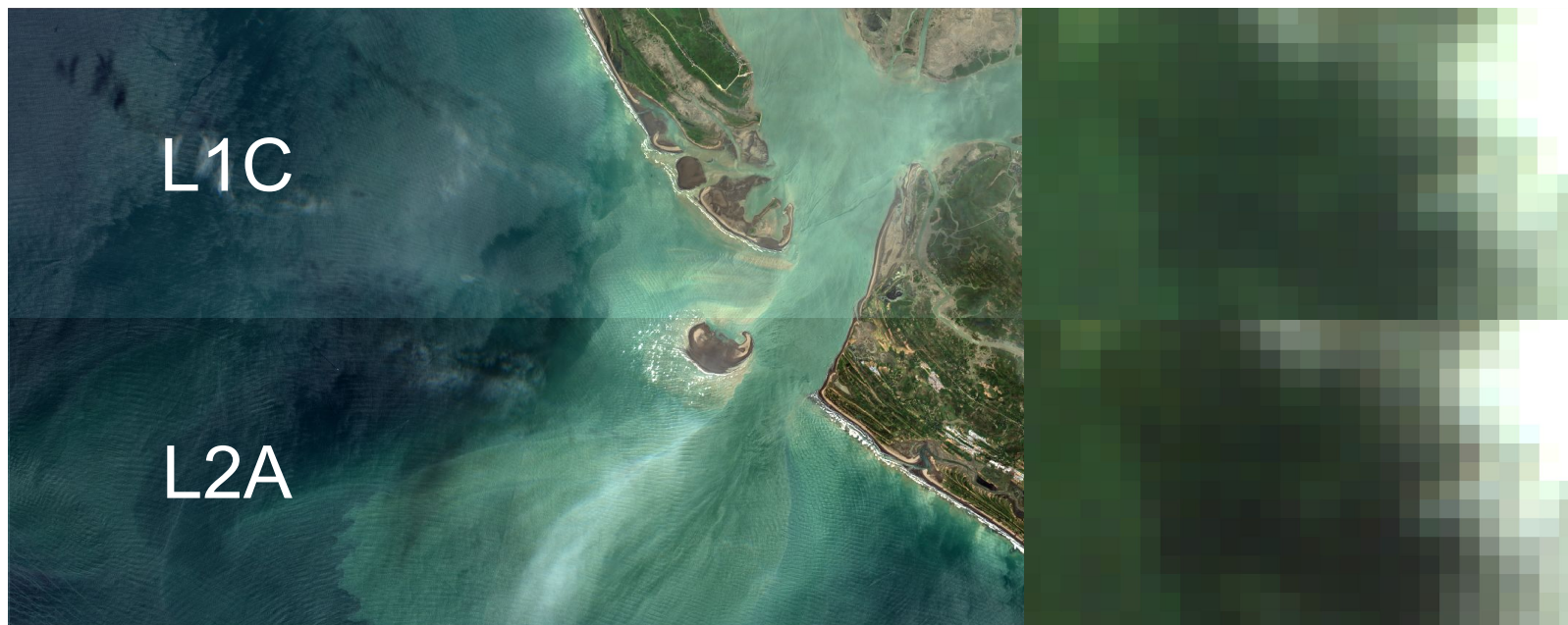
Data Acquisition

- discovery through SciHub API
(copernicus open access hub)
- download using `sentinelSAT`
- organizing: one directory for each run
containing all scenes



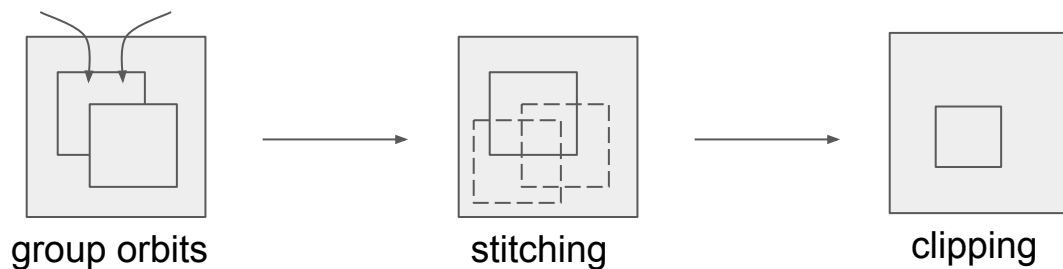
Preprocessing

- Atmospheric Correction via Sen2Cor 2.5.5 by ESA



Preprocessing

- Subset bands & select resolution
- Tile stitching: group images by orbit
- Clip to AoI
- Reproject (EPSG 4326)
- output single GeoTIFF



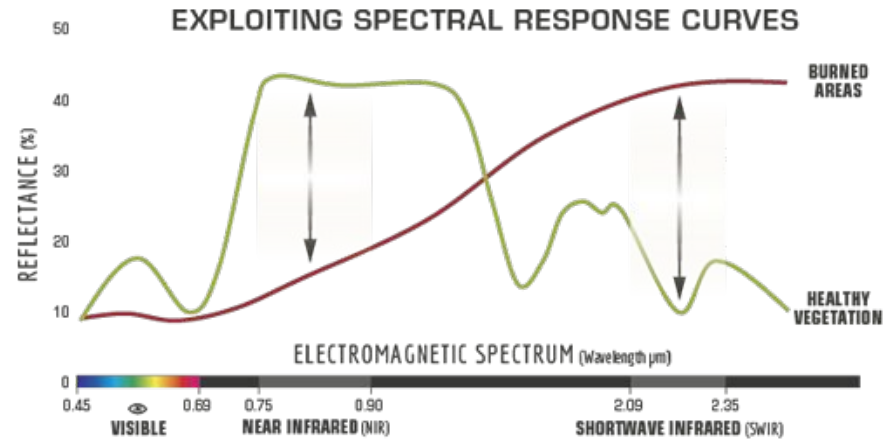
Change Detection



Difference Normalized Burn Ratio

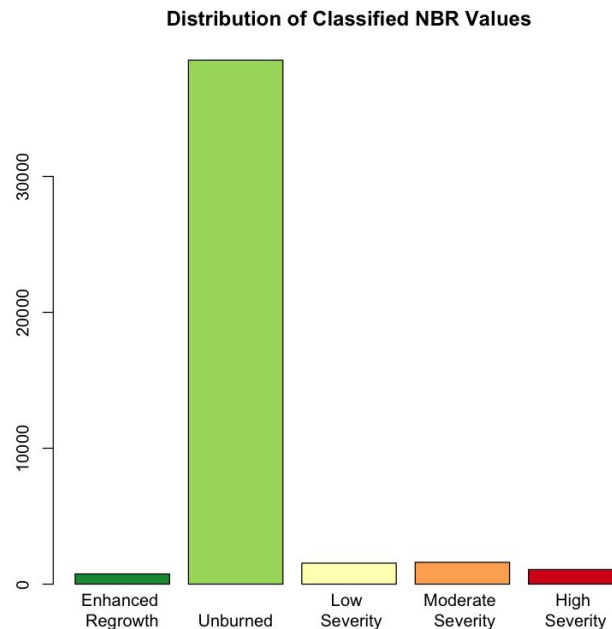
- Identifies Burned Areas
- Near Infrared (Band 8)
- Shortwave Infrared (Band 12)
- $NBR = (NIR - SWIR) / (NIR + SWIR)$
- $dNBR = \text{Prefire NBR} - \text{Postfire NBR}$

=> only vegetation



Severity Level

- severity is a qualitative term
- hard to determine
- different for every use-case
- fixed vs. dynamic classifications



Burned Area Index for Sentinel-2

- detect burned areas at 20 m spatial resolution [1]
- design of a processor developed to perform post-fire mapping using Sentinel-2 data [1]

Formula [1]:

$$BAIS2 = \left(1 - \sqrt{\frac{B06 * B07 * B8A}{B4}} \right) * \left(\frac{B12 - B8A}{\sqrt{B12 + B8A}} + 1 \right)$$

Water Detection

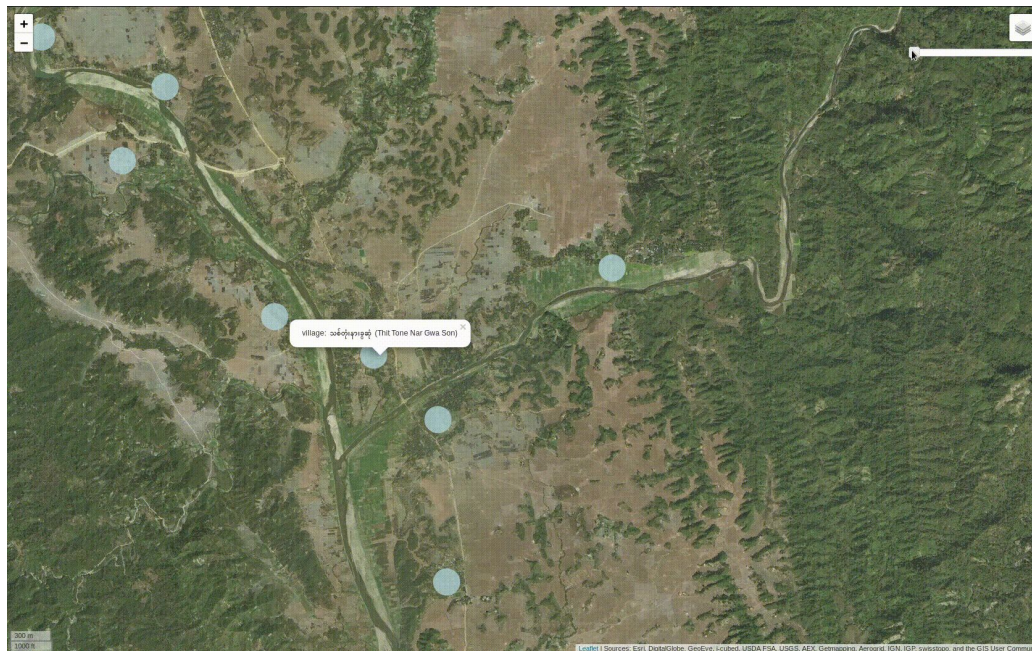
- Detects water pixels
- used to enhance change detection
- water pixels are subtracted from output of change detection

Formula[1] :

$$WP = \left(\frac{(B8A + B11 + B12) - (B01 + B02 + B03)}{(B8A + B11 + B12) + (B01 + B02 + B03)} \right) < 0$$

Visualization

- Interactive webmap, for
 - each change detection method
 - unprocessed I2a images
- Automatically generated



Results // Demo

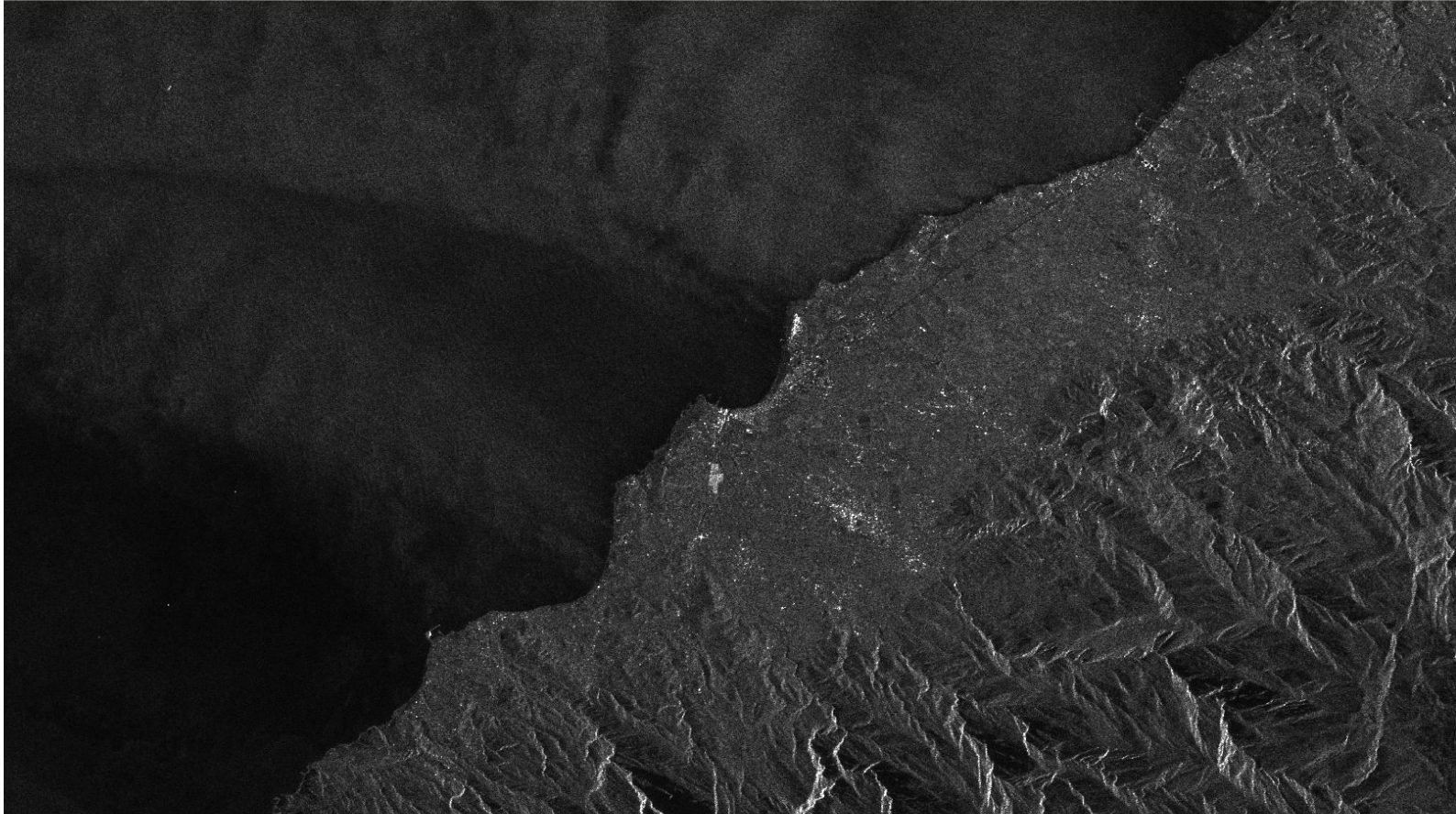
Myanmar | Dec 2017 - Feb 2018 | Cloudcoverage 60%

[Screencast Pipeline](#)

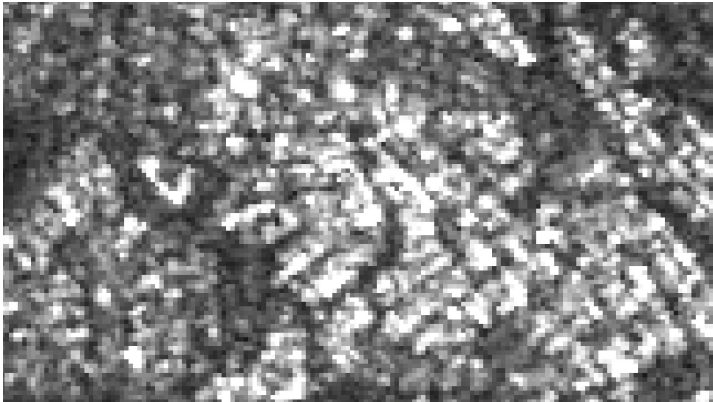
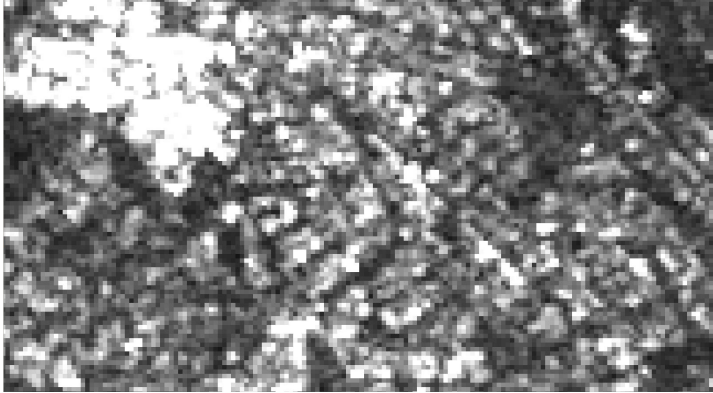
Approaches & future work

- Radar imagery
- Aol optimization
- Visualization
- Cloud mask
- Thermal imagery
- Village detection

Radar imagery: water detection

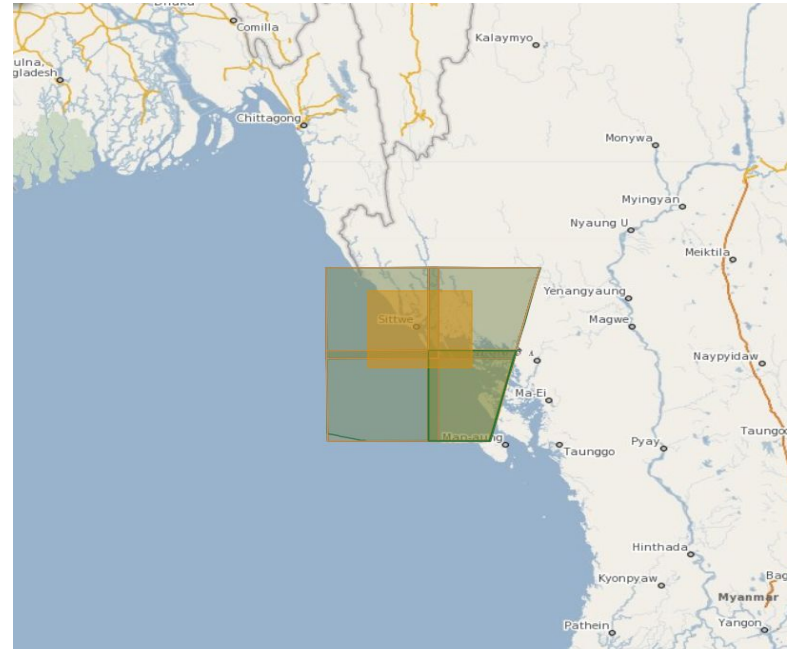
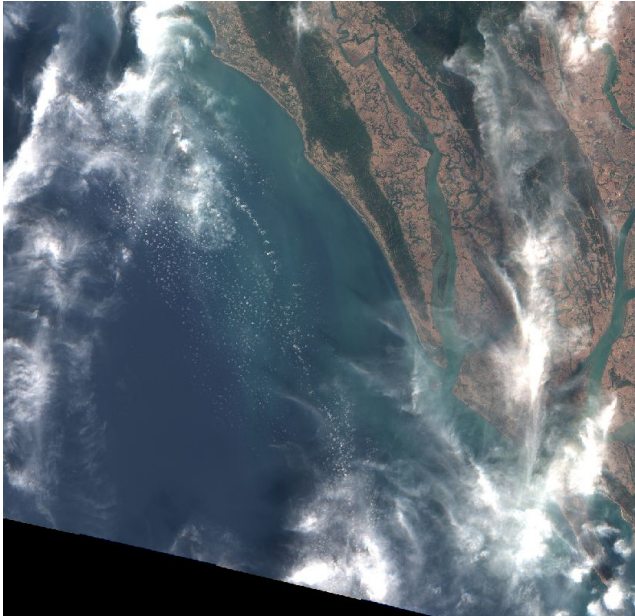


Radar imagery: physical destruction



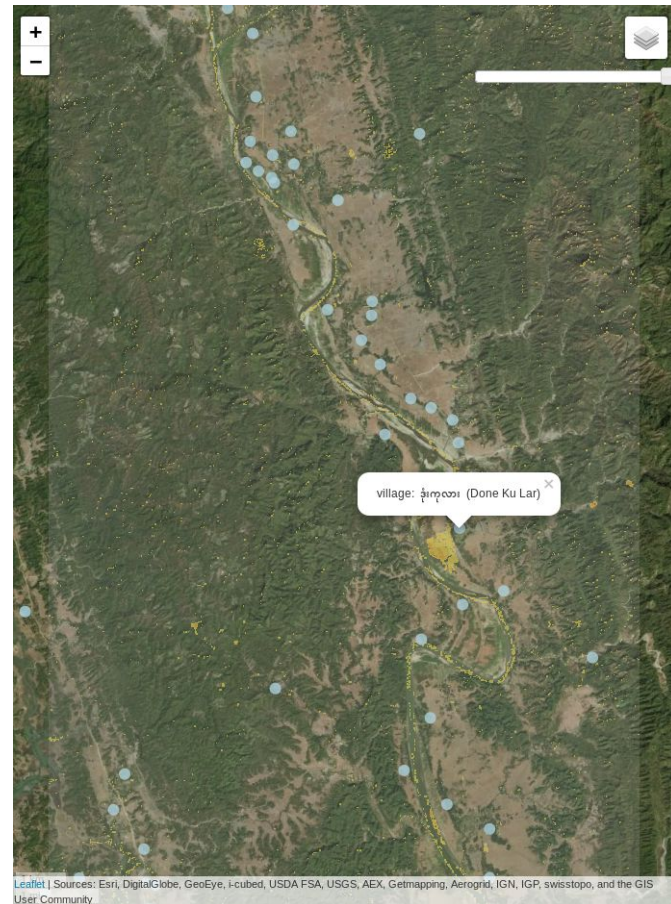
Approaches: Aoi

- apply cloud % to Aoi only, not to whole tiles
- skip tiles with low overlap with aoi



Approaches: Visualization

- interactive visualization implemented
- possible improvement:
 - Animated image time series with changes highlighted
 - Diagrams showing amount of destruction by time



Approaches: Cloud Detection & thermal imagery

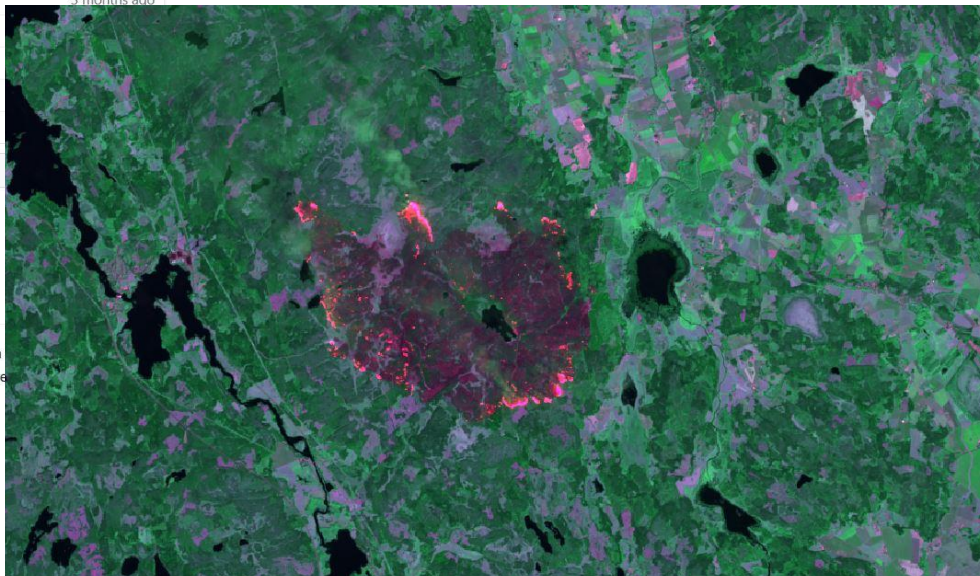
MANIFEST.in	setup update, added travis, license, manifest, pylintrc, requirements...	4 months ago
Makefile	setup update, added travis, license, manifest, pylintrc, requirements...	4 months ago
README.md	Version 1.2.1 - A few minor updates, added support for Python3.7	3 months ago
pylintrc	setup update, added travis, license, manifest, pylintrc, requirements...	
requirements-dev.txt	setup update, added travis, license, manifest, pylintrc, requirements...	
requirements.txt	version 1.2.2 - improved unittests, imported from sentinelhub-py	
setup.py	setup update, added travis, license, manifest, pylintrc, requirements...	

README.md

pypi package 1.2.2 python 3.5 | 3.6 | 3.7 build passing downloads 8k downloads/month 1k

Sentinel Hub's cloud detector for Sentinel-2 imagery

The `s2cloudless` Python package provides automated cloud detection in Sentinel-2 imagery. The classification is based on a *single-scene pixel-based cloud detector* developed by Sentinel Hub's research team and is described in more detail in the [Sentinel Hub Cloud Detector](#) paper.

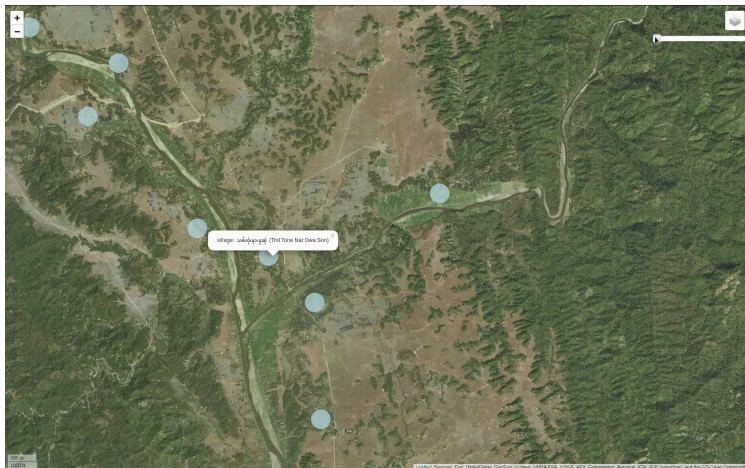


Approaches & future work

- Sentinel 1 Radar → destructed buildings, advanced water detection ✓
- Aol → cloud coverage, low overlapping area ✓
- Visualization → animated, background data
- Cloud Mask → less false positives ✓
- Thermal imagery → detection of current fires ✓
- Village detection → join results with related datasets

Conclusion

- Complete processing pipeline
 - extensible
 - dockerized



- Change detection algorithms
 - dNBR
 - BAIS2
- Further improvements through
 - better change detection parametrization
 - adding change detection algorithms
 - extended visualization of results

Questions



References

- [1] Federico Filipponi: BAIS2: Burned Area Index for Sentinel-2
- [2] <https://www.dabangasudan.org/en/all-news/article/attacks-on-jebel-marra-continue-three-darfur-students-killed>
- [3] <https://www.abc.net.au/news/2017-11-23/myanmar-burning-village/9184026>
- [4] <https://github.com/sentinel-hub/sentinel2-cloud-detector>
- [5] <http://www.digital-geography.com/forest-fire-tracking-landsat-8/>
- [6] <http://www.sattec.org>