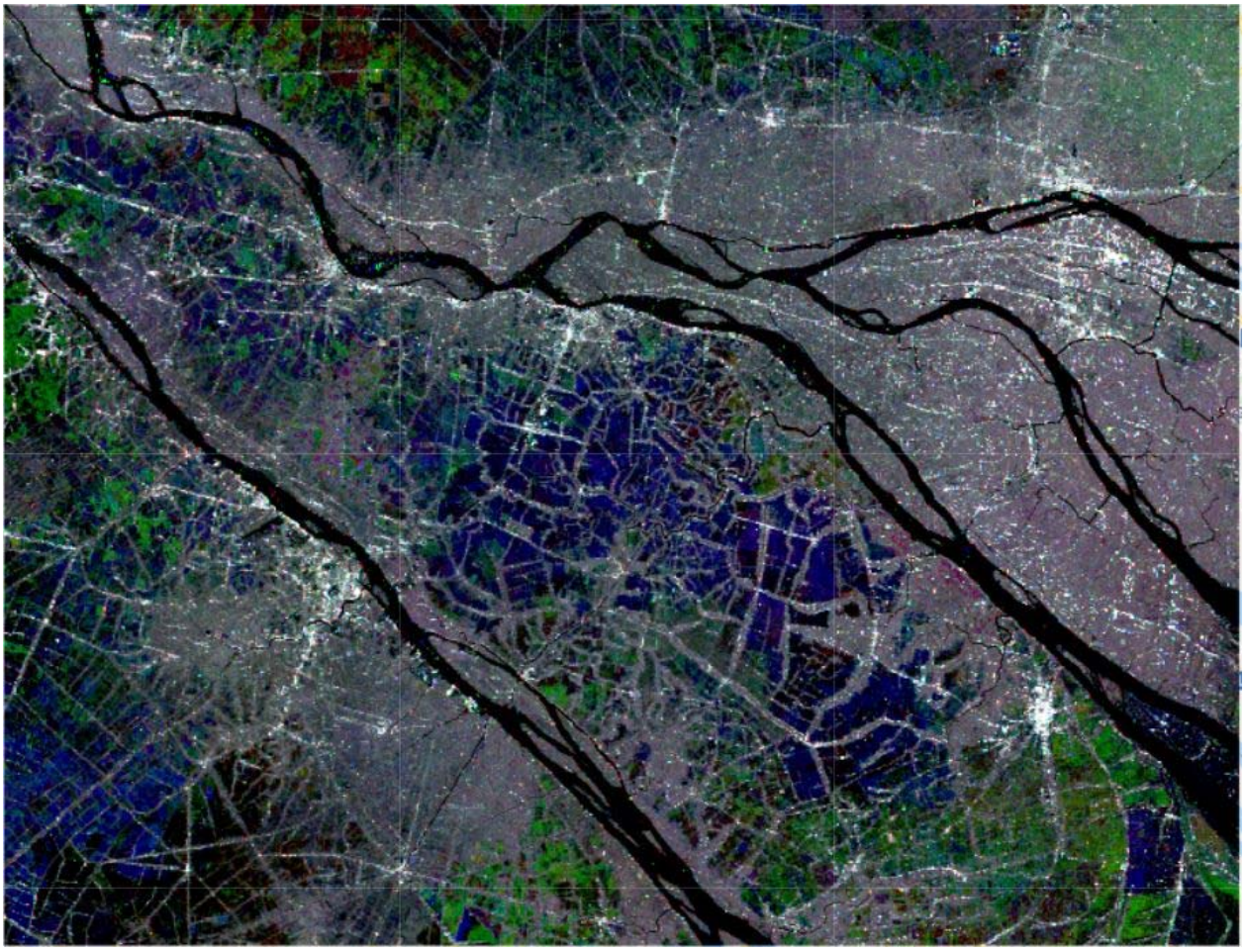
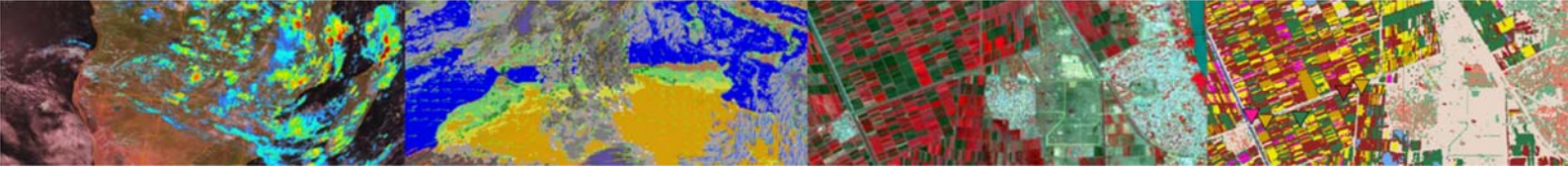
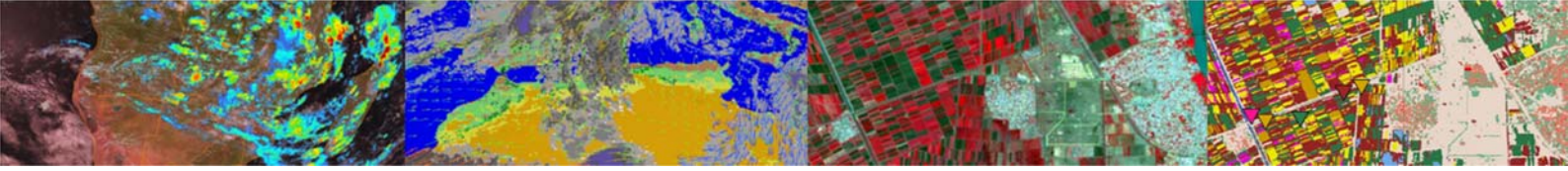


Application of Sentinel for Land and Water
Case studies Vietnam

Editor: Ben Maathuis, 20-08-2021



Contents



1. Assessment of flood extent in Quang Binh province, Vietnam, using Sentinel-1 images and Google Earth Engine platform

Vuong Tai Chi (chivuongtai@gmail.com)

Being the Central part of Vietnam, Quang Binh province has a diversity of water resources with a dense river network. However, in October 2020, this province has suffered the most severe flood event in many decades. Therefore, assessment of flood extent in Quang Binh province using Sentinel-1 images is selected (Figure 1).

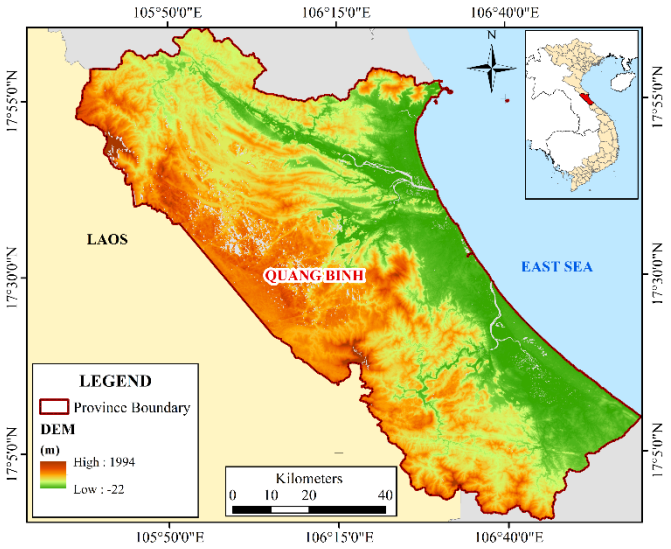


Figure 1. Administration map of Quang Binh province

Among the techniques introduced within the course, I employed Java Script to analyze Sentinel-1 images on the cloud-based platform, Google Earth Engine. This approach helps me reduce significant time for downloading data and deal with the issue of computer's capacity for storing satellite images. On the other hand, Sentinel-1 images were selected as the target data because they are independent on weather conditions, especially cloud that usually occurs during flood events and becomes the hug obstacle if analyzing images from Optical Satellite recordings.

In order to assess the flood extent in Quang Binh province, there were 5 steps conducted as shown in Figure 2.

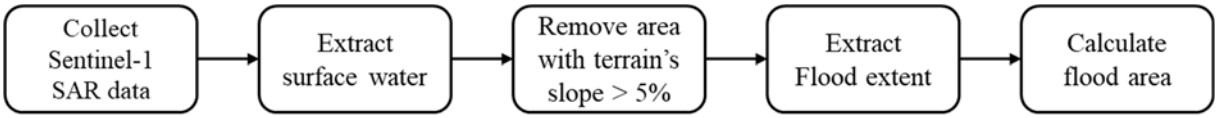


Figure 2. The flow chart of flood extent assessment

There were 4 images collected in this report including 2 images during the flood event in October 2020 and 2 reference images in August 2020, when the flood event had not occurred. These images were filtered with VH (Vertical transmit – Horizontal receive) polarization, descending pass direction, and 10m resolution. Before further processing, the selected images were smoothed to reduce the radar speckle, or the 'grainy' appearance in the image.

For surface water extraction, pixel values of during-flood-event images were divided by the before one, then the threshold of 1.35 was chosen to produce the temporal water layer. The generated temporary water layer includes floodwater and the water resulting from radar error on mountainous area, where its shadow is also assigned to low values, corresponding to the values of water. In this context, the areas having a slope of greater than 5% were removed based on the DEM layer from the HydroSHEDS data, also available in GEE. This procedure helps ignoring the non–correct pixels and retrieving only the floodwater layer. As can be seen from the method applied, within the procedure of floodwater extraction the removal of permanent water is not conducted. The reason is that permanent water is included in both before and during-flood-event images; therefore, when taking the ratio between them and choosing a threshold greater than 1, permanent water is ignored by itself.

For flood area calculation, the area of each pixel being detected as floodwater was calculated. Then the sum of all floodwater pixel areas gives us the result of flood area in the whole region of interest. Figure 3 shows that Quang Binh province flooded area is 19,124 ha. and the flooded area mainly occurs along the rivers.

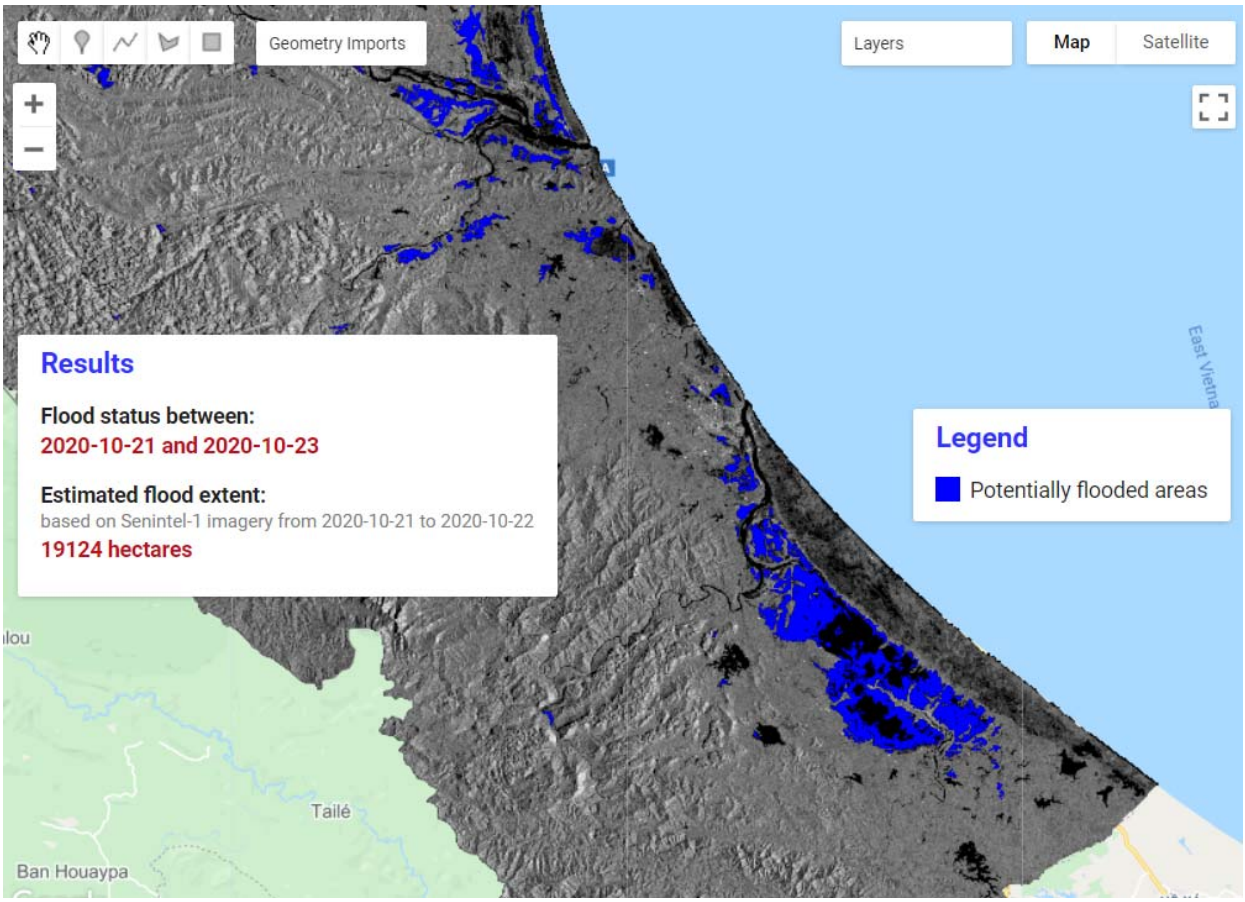
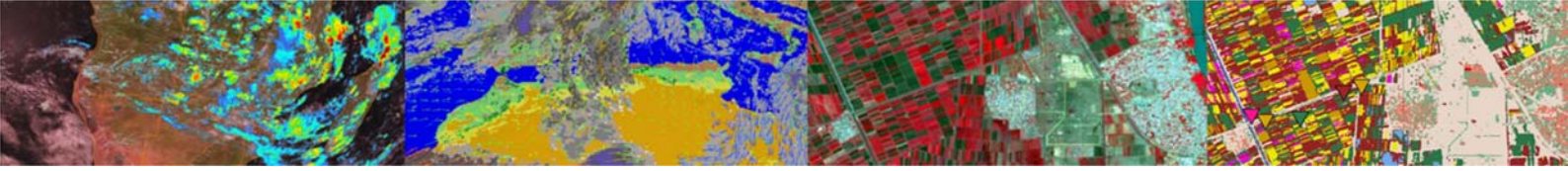
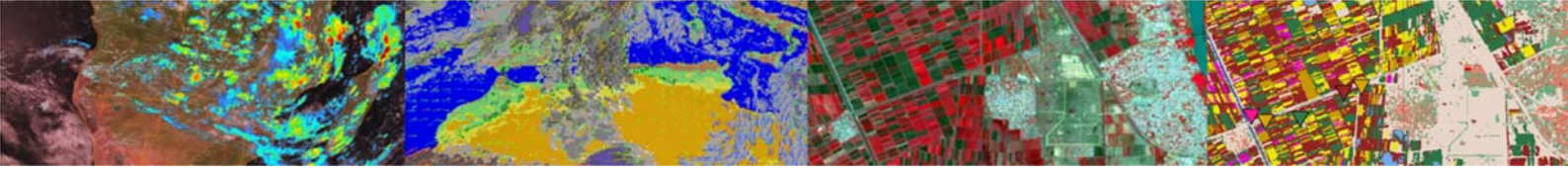


Figure 3. The flood extent in Quang Binh province

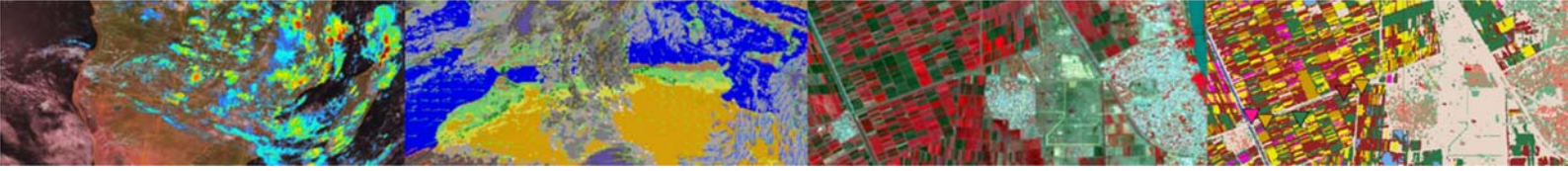


Annex: Script used in the report (Java)

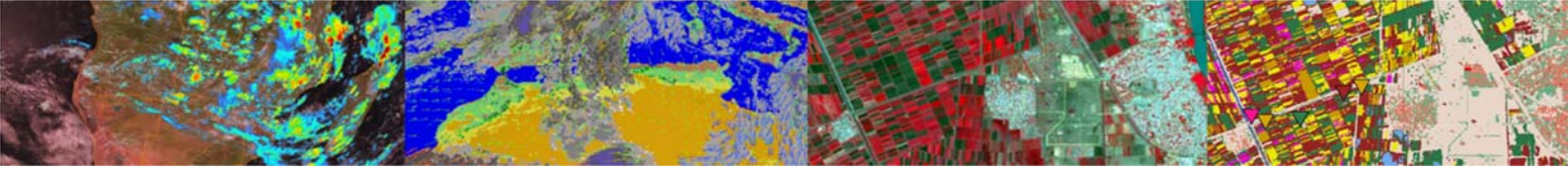
```
//----------------------IMAGE SELECTION-----------------------

var table = ee.FeatureCollection("users/chivuongtai/QuangBinh_province");
var aoi = ee.FeatureCollection(table); // area of interest
var z = 9; // zoom level

//BEFORE the flood.
var before_start = '2020-08-23';
var before_end ='2020-08-29';

//AFTER the flood.
var after_start ='2020-10-21';
var after_end ='2020-10-23';

//SETTING
var polarization = "VH";
var pass_direction = "DESCENDING"; //or 'ASCENDING'
var difference_threshold = 1.35;

//FITLTER SENTINEL-1 DATA
var collection= ee.ImageCollection('COPERNICUS/S1_GRD')
  .filter(ee.Filter.eq('instrumentMode','IW'))
  .filter(ee.Filter.listContains('transmitterReceiverPolarisation', polarization))
  .filter(ee.Filter.eq('orbitProperties_pass',pass_direction))
  .filter(ee.Filter.eq('resolution_meters',10))
  //.filter(ee.Filter.eq('relativeOrbitNumber_start',relative_orbit ))
  .filterBounds(aoi)
  .select(polarization);

var before_collection = collection.filterDate(before_start, before_end);
var after_collection = collection.filterDate(after_start,after_end);

//EXTRACT DATE & PRINT TO CONSOLE
function dates(imgcol){
       var range = imgcol.reduceColumns(ee.Reducer.minMax(), ["system:time_start"]);
       var printed = ee.String('from ')
         .cat(ee.Date(range.get('min')).format('YYYY-MM-dd'))
         .cat(' to ').cat(ee.Date(range.get('max')).format('YYYY-MM-dd'));
       return printed;}

var before_count = before_collection.size();
     print(ee.String('Tiles selected: Before Flood ').cat('(').cat(before_count).cat(' images)'),
       dates(before_collection), before_collection);
var after_count = before_collection.size();
     print(ee.String('Tiles selected: After Flood ').cat('(').cat(after_count).cat(' images)'),
       dates(after_collection), after_collection);

//CLIP TO AOI
var before = before_collection.mosaic().clip(aoi);
var after = after_collection.mosaic().clip(aoi);

//REDUCE RADAR SPECKLE
var smoothing_radius = 50;
var before_filtered = before.focal_mean(smoothing_radius, 'circle', 'meters');
var after_filtered = after.focal_mean(smoothing_radius, 'circle', 'meters');
```



```
//----------------------DETECT AND CALCULATE FLOOD EXTENT-----------------------

// TAKE RATIO OF IMAGES AFTER & BEFORE FLOOD EVENT
var difference = after_filtered.divide(before_filtered);
var threshold = difference_threshold;
var difference_binary = difference.gt(threshold);              //gt = greater than

// REDUCE NOISE
var flooded = difference_binary.updateMask(difference_binary);
var connections = flooded.connectedPixelCount();
var flooded = difference_binary.updateMask(connections.gte(8));

// FILTER SLOPE>5% (using DEM)
var DEM = ee.Image('WWF/HydroSHEDS/03VFDEM');
var terrain = ee.Algorithms.Terrain(DEM);
var slope = terrain.select('slope');
var flooded = flooded.updateMask(slope.lt(5)); //lt = less than

//CALCULATE FLOOD AREA
var flood_pixelarea = flooded.select(polarization).multiply(ee.Image.pixelArea());
var flood_stats = flood_pixelarea.reduceRegion({
  reducer: ee.Reducer.sum(),
  geometry: aoi,
  scale: 10, // native resolution
  maxPixels: 1e9,
  bestEffort: true
  });
var flood_area_ha = flood_stats.getNumber(polarization).divide(10000).round();

//-------------------------------SHOW RESULT LAYERS---------------------------------

Map.centerObject(aoi,z);//Results appearance will focus on aoi
Map.addLayer(before_filtered, {min:-25,max:0}, 'Before Flood',0);
Map.addLayer(after_filtered, {min:-25,max:0}, 'After Flood',1);
Map.addLayer(difference,{min:0,max:2},"Difference Layer",0);
Map.addLayer(flooded,{palette:"0000FF"},'Flooded areas');

//-------------------------------EXPORT RESULT LAYERS--------------------------------

// Export flooded area as TIFF file
Export.image.toDrive({
  image: flooded,
  scale: 10,
  description: 'FloodExtentTIFF',
  region: aoi,
  maxPixels: 1e10
});
// Export flooded area as shapefile (SHP) (2 steps)
// 1. Convert flood raster to polygons
var flooded_vec = flooded.reduceToVectors({
  scale: 10,
  geometryType:'polygon',
  geometry: aoi,
  eightConnected: false,
  bestEffort:true,
  tileScale:2,
});
// 2. Export flood polygons as shape-file
Export.table.toDrive({
  collection:flooded_vec,
  description:'FloodExtentSHP',
```



```
  fileFormat:'SHP',
  folder: 'QBinh'
});

//----------------------------SHOW RESULT STATISTICS-----------------------------

// set position of panel where the results will be displayed
var results = ui.Panel({style: {position: 'middle-left',padding: '8px 15px',width: '350px'}});

//Prepare the visualtization parameters of the labels
var textVis = {'margin':'0px 8px 2px 0px','fontWeight':'bold'};
var numberVIS = {'margin':'0px 0px 15px 0px','color':'bf0f19','fontWeight':'bold'};
var subTextVis = {'margin':'0px 0px 2px 0px','fontSize':'12px','color':'grey'};
var titleTextVis = {'margin':'0px 0px 15px 0px','fontSize': '18px','font-weight':'bold','color': '3333ff'};

// Create lables of the results
// Titel and time period
var title = ui.Label('Results', titleTextVis);
var text1 = ui.Label('Flood status between:',textVis);
var number1 = ui.Label(after_start.concat(" and ",after_end),numberVIS);

// Estimated flood extent
var text2 = ui.Label('Estimated flood extent:',textVis);
var text2_2 = ui.Label('Please wait...',subTextVis);
dates(after_collection).evaluate(function(val){text2_2.setValue('based on Senintel-1 imagery '+val)});
var number2 = ui.Label('Please wait...',numberVIS);
flood_area_ha.evaluate(function(val){number2.setValue(val+' hectares')}),numberVIS;

// Add the labels to the panel
results.add(ui.Panel([title,text1,number1,text2,text2_2,number2]));
Map.add(results);

//--------------------------------SHOW RESULT LEGEND-------------------------------

// Create legend
var legend = ui.Panel({style: {position: 'middle-right',padding: '8px 15px',}});
var legendTitle = ui.Label('Legend',titleTextVis);
legend.add(legendTitle);

// Creates and styles 1 row of the legend
var makeRow = function(color, name) {
      var colorBox = ui.Label({style: {backgroundColor: color,padding: '8px',margin: '0 0 4px 0'}});
      var description = ui.Label({value: name, style: {margin: '0 0 4px 6px'}});
      return ui.Panel({widgets: [colorBox, description],layout: ui.Panel.Layout.Flow('horizontal')});
};

//  Palette with the colors
var palette =['#0000FF'];
var names = ['Potentially flooded areas'];
legend.add(makeRow(palette, names));
Map.add(legend);
```



2. Suspended sediment dynamics in the Mekong Delta estuaries using Sentinel

Doan Van Binh (binhdv0708vl@gmail.com)

Introduction

The flow in the Mekong Delta is highly dynamic due to bidirectional flows resulting from the interaction between the riverine flow from the Mekong River and tidal regimes from the East Vietnam Sea. Understanding sediment dynamics in the delta is of strategic importance for linking with the evolution of the delta front and bed morphology. Monitoring suspended sediment in the delta is extremely sparse; the data are long-term available at some main hydrological stations. Moreover, the temporal resolution of the monitored data is coarse, at daily basic. To fully understanding the spatiotemporal dynamics of the suspended sediment in the delta need innovative method, and therefore, satellite imagery can fulfill this requirement. Among satellite sources, Sentinel 2 and 3 with highly spatial resolution and revisited frequency, respectively, are employed.

Objective

The objective is to examine the spatiotemporal variations of the suspended sediment in the Mekong Delta estuaries. The specific objectives are:

- To use Sentinel 3 TSM-NN for understanding spatiotemporal variations of sediment concentration. Sentinel EO4SD Toolbox and QGIS are used.
- To establish the relationship (equation) between the reflectance of Sentinel 2 and the observed suspended sediment data. QGIS is used.
- To estimate the sediment concentration (map) by using the established equation. QGIS is used.
- To examine to spatiotemporal variations of the sediment concentration
- To compare the results between Sentinel 3 TSM-NN and Sentinel 2

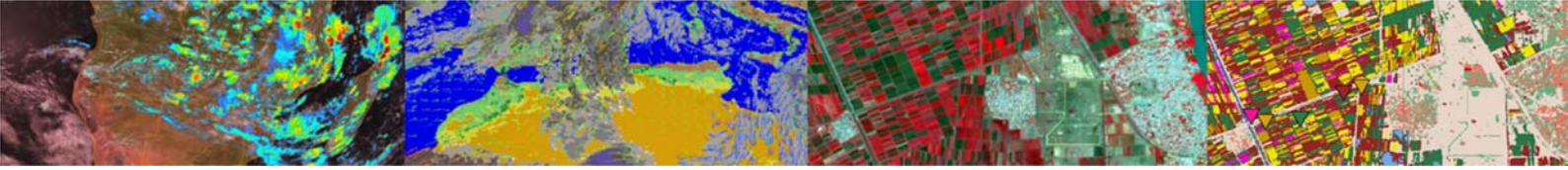
Results and discussion

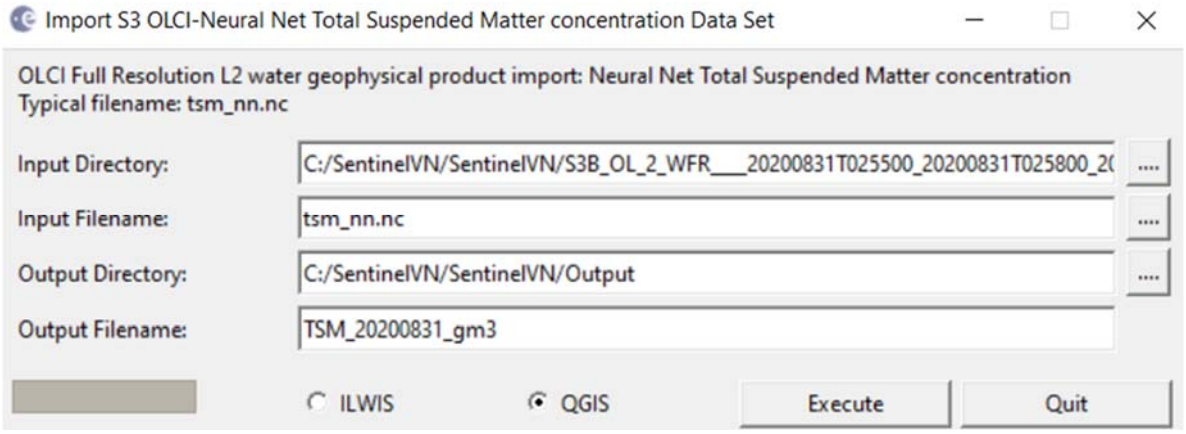
Sediment dynamics using Sentinel 3

Four images of Sentinel 3 level 2 OLCI (Ocean and Land Colour Instrument) in full resolution (FR) are downloaded from https://coda.eumetsat.int/#/home. The images' names are:

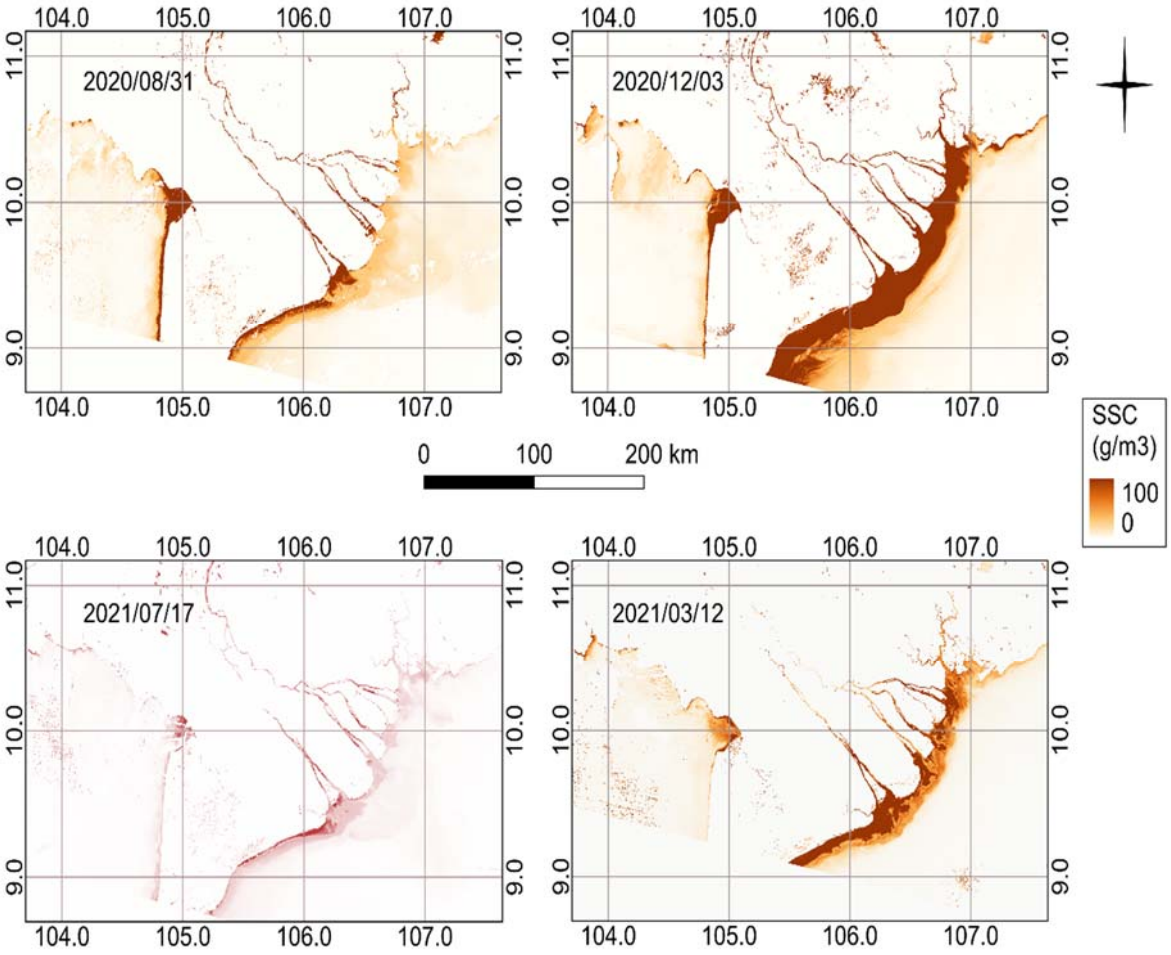
- S3A_OL_2_WFR____20201203T025656_20201203T025956_20201204T104901_0179_065_360_2700_MAR_O_NT_002.zip
- S3B_OL_2_WFR____20200831T025500_20200831T025800_20200901T120441_0179_043_032_2700_MAR_O_NT_002.zip
- S3B_OL_2_WFR____20210717T025854_20210717T030154_20210717T050523_0179_054_360_2700_MAR_O_NR_003.zip
- S3B_OL_2_WFR____20210312T025116_20210312T025416_20210313T145931_0180_050_089_2700_MAR_O_NT_003.zip

These products were atmospherically corrected to the bottom-of-atmosphere. The total suspended matter (TSM_NN) is used. The data is opened by Sentinel EO4SD Toolbox and then transferred to QGIS software.

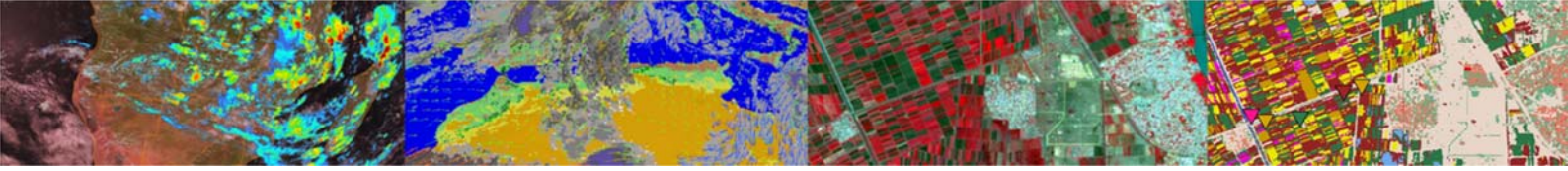




In QGIS, the newly generated TSM files are opened and displayed in a manner that all images have the same data range for comparison. Seasonal variations of the suspended sediment are compared in the following figure.



It can be seen that 3 out of 4 images have a maximum of 100 g/m^3 of the suspended matter in both flood and dry seasons, which is unreasonable. Although the riverbed in the dry season maybe re-suspended due to wind and wave, the sediment in the dry season is much higher than during the flood season as indicated in the figure above and is considered unreliable. Therefore, total suspended matter from Sentinel 3 in the Mekong Delta has to be validated using *in-situ* measurements. A regression analysis should be performed between TSM-NN and measured data from which the adjusted sediment is mapped accordingly, e.g. using the Raster Calculator in QGIS.



Sediment dynamics using Sentinel 2

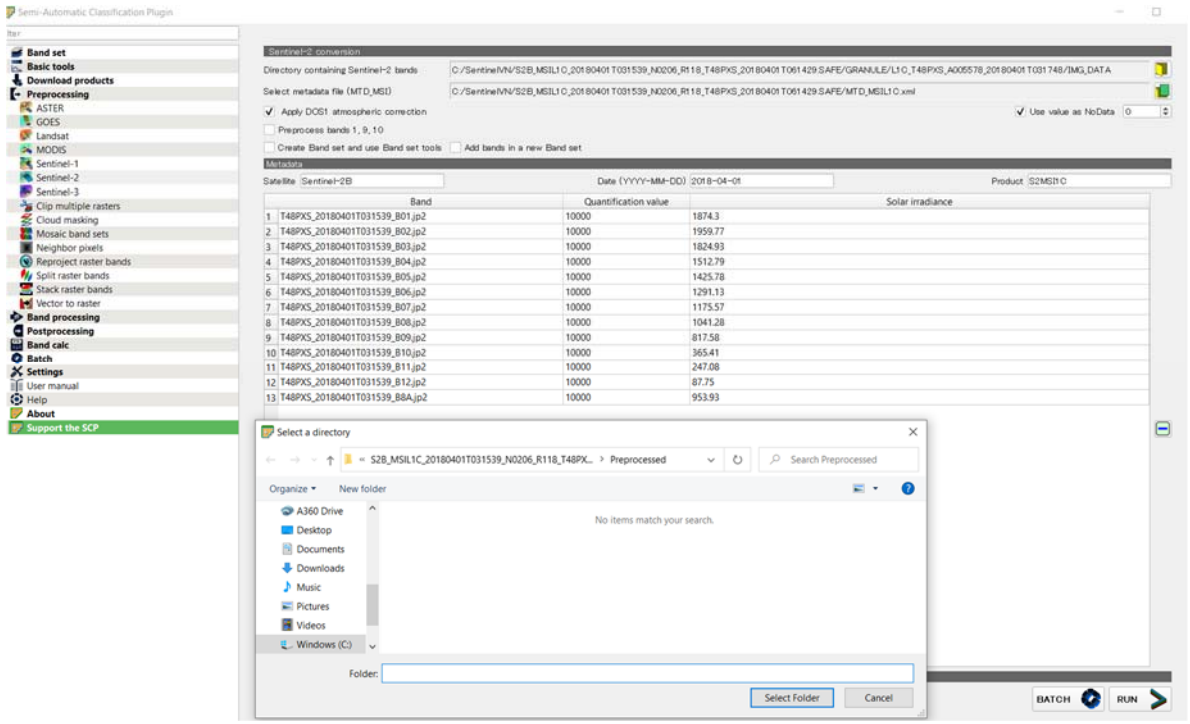
The data were downloaded from https://scihub.copernicus.eu/dhus/#/home

Establishing regression equation using Sentinel 2 level 1

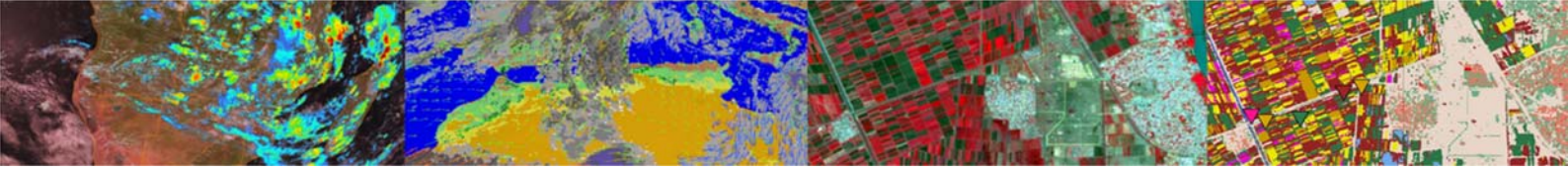
The *in-situ* measurement was conducted on 6-9 April 2018; however, Sentinel 2 in this period is not available because of cloud cover. Therefore, the data of 1 April 2018, closest to the observed data, is used. The data downloaded is level 1, without atmospheric correction and the values are in radiances.
The file name is S2B_MSIL1C_20180401T031539_N0206_R118_T48PXS_20180401T061429.zip. In QGIS, the red band (Band 4 – B04) is opened to correlate with the observed data.

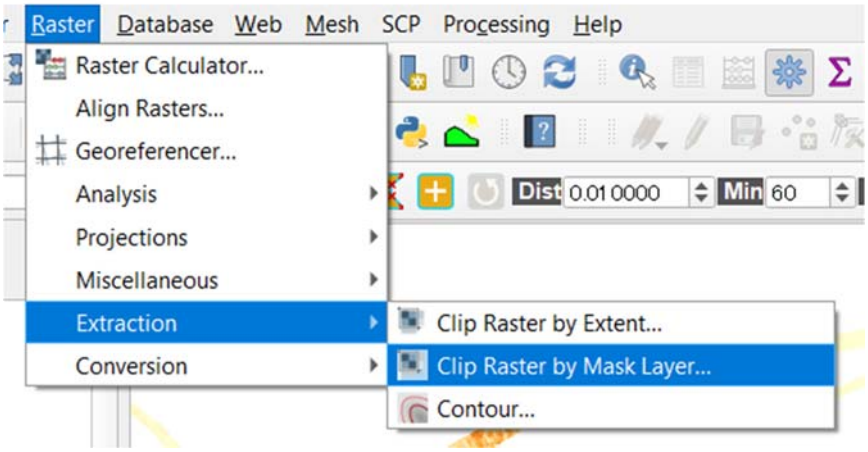
The path is C:\SentinelVN\S2B_MSIL1C_20180401T031539_N0206_R118_T48PXS_20180401T061429.SAFE\GRANULE\L1C_T48PXS_A005578_20180401T031748\IMG_DATA.

First, the image is atmospherically corrected using Preprocessing option in the Semi-Automatic Classification Plugin in QGIS as in the following figure

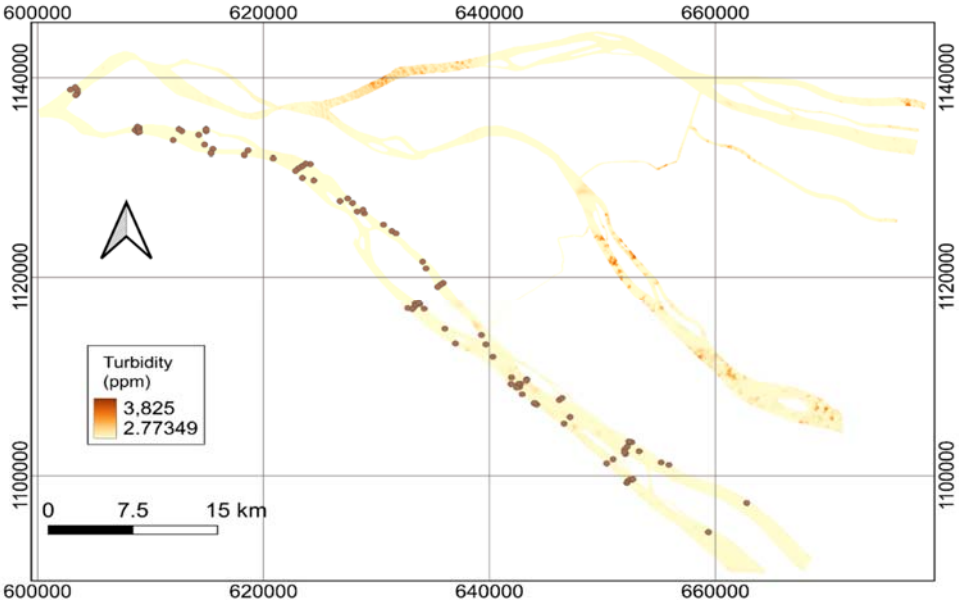


Then, the corrected image is converted from radiance to reflectance by dividing by 10,000 using Raster Calculator in QGIS. A polygon shapefile of the study area is plotted into the map to mask the big Sentinel 2 image to the river shapefile only where the measurement was conducted. Using Raster Extraction to mask the map as

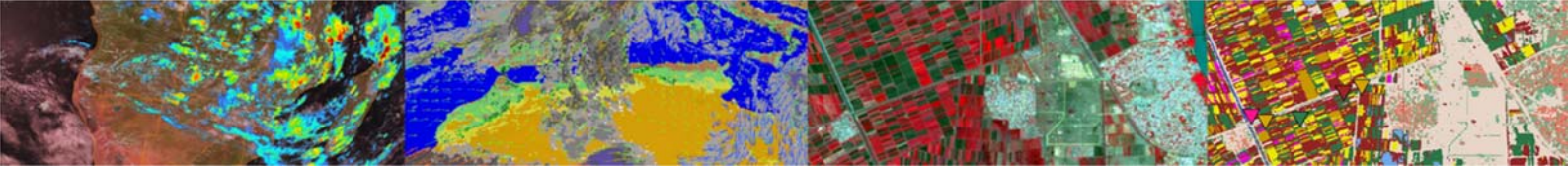


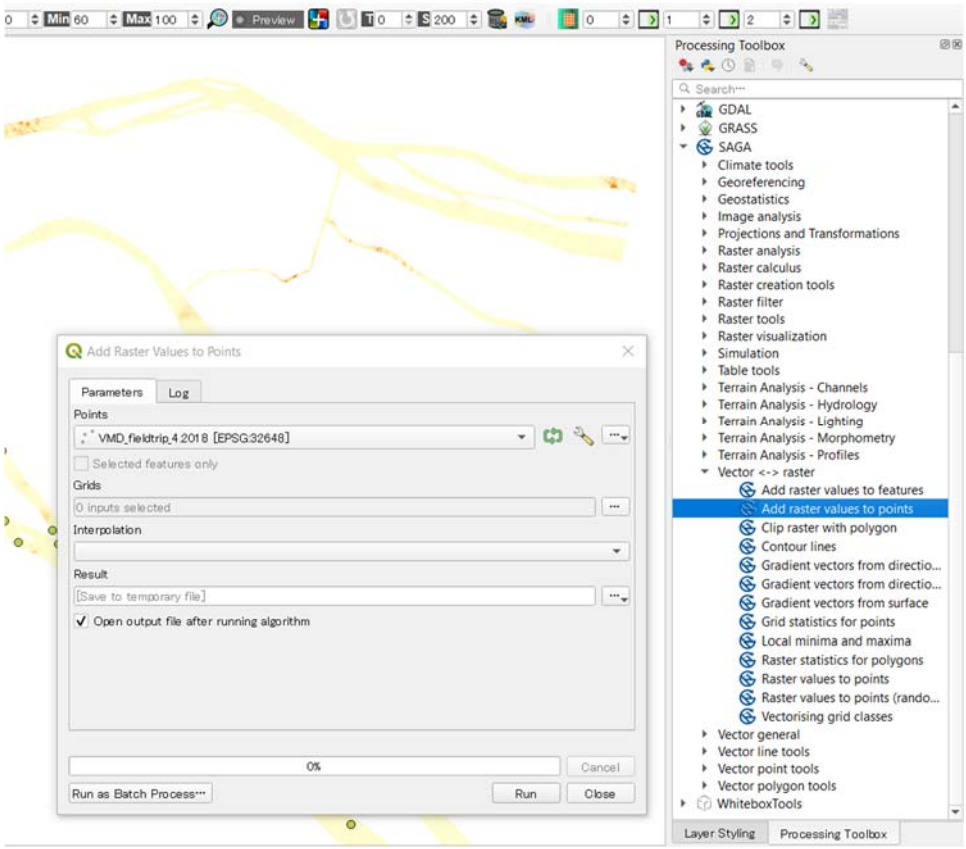


Next, a point shapefile of the locations where the suspended sediment concentrations were measured during the field trip is plotted to the map in QGIS.

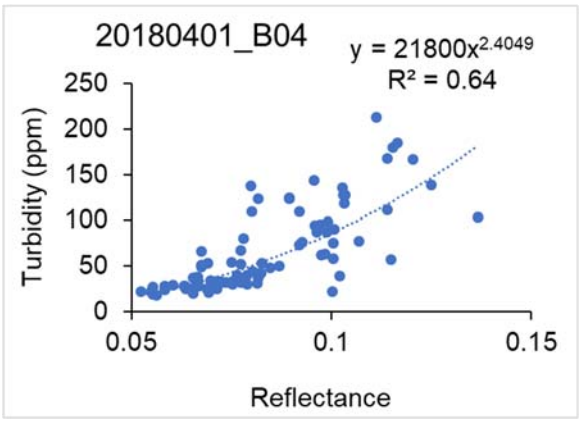


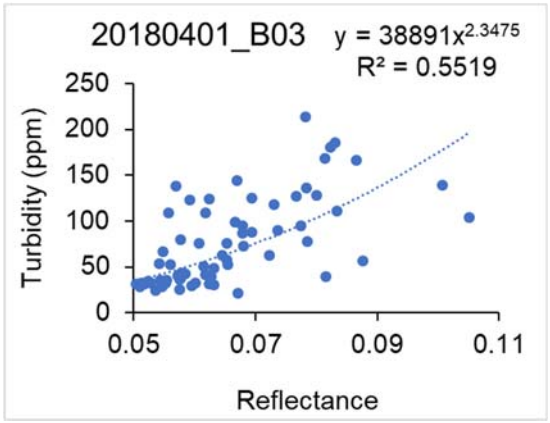
This shapefile is used to extract the pixel value of Sentinel image to the point measurement for the purpose of establishing the regression equation.





Procedure on https://www.youtube.com/watch?v=Mx_M3fB3f6Q).
The extracted reflectance values at the pixels having in-situ data is plotted versus the observed values using Excel. The regression equations are established from Band 4 (red band), Band 3 (green band) and Band 8 (Near Infrared band). The procedure to process Band 3 and Band 8 is the same with Band 4, following the over steps.

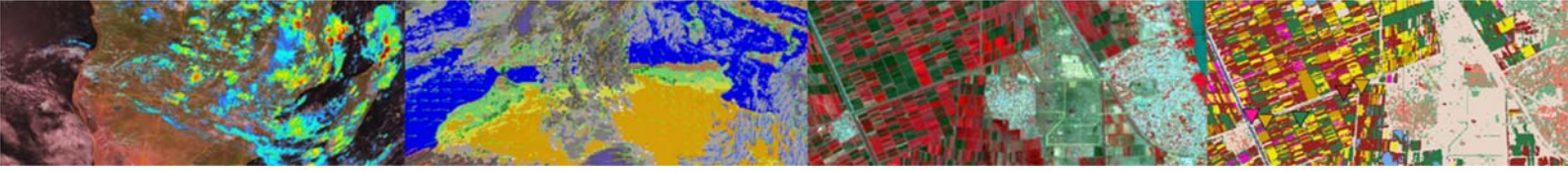




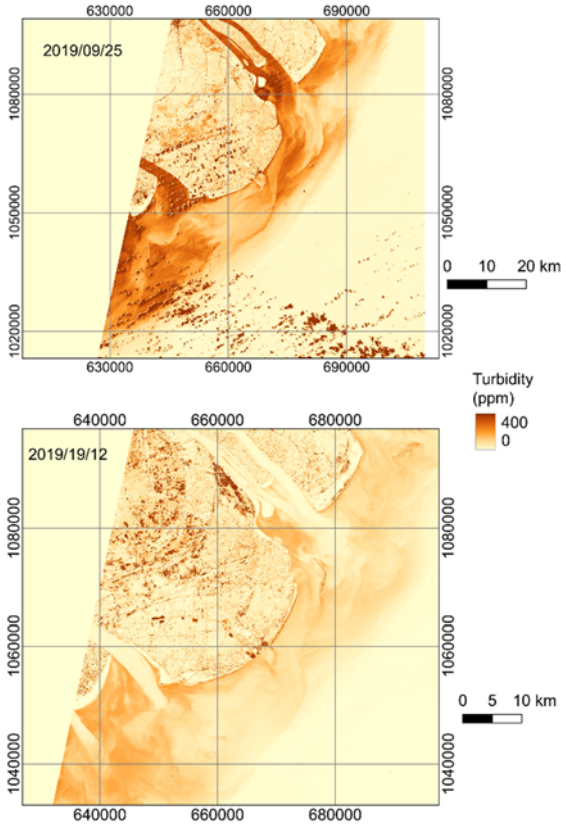
From these two figures, it can be said that Band 4 is likely better than Band 3 in suspended sediment estimation in the Mekong Delta. Therefore, the corresponding regression equation using Band 4 is used to map the suspended sediment concentration in the study area.

Seasonal variations of suspended sediment concentration using Sentinel 2 level 2

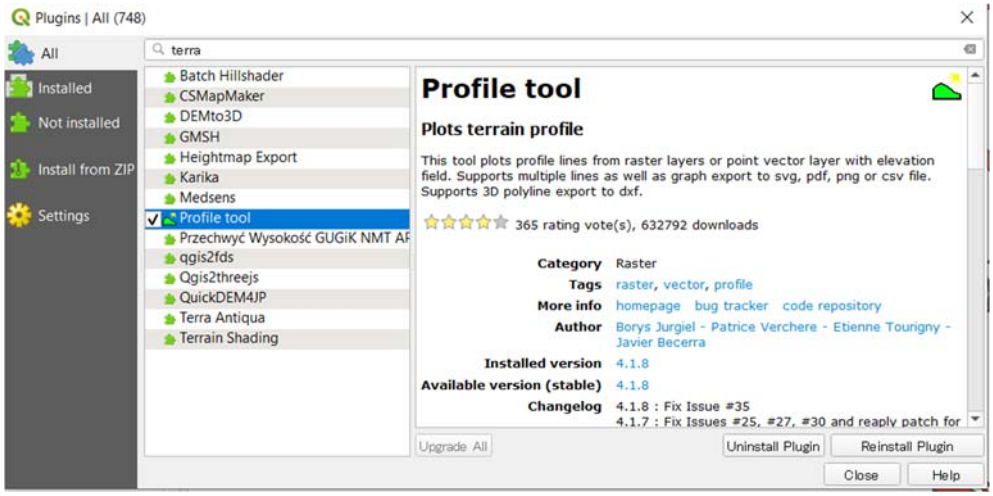
Two images of Sentinel 2 level 2A (atmospheric correction and the pixel values are in reflectance already) are used; one in the flood and one in the dry season:
S2A_MSIL2A_20190925T030541_N0213_R075_T48PXR_20190925T091801.zip
S2B_MSIL2A_20191219T031129_N0213_R075_T48PXR_20191219T064247.zip

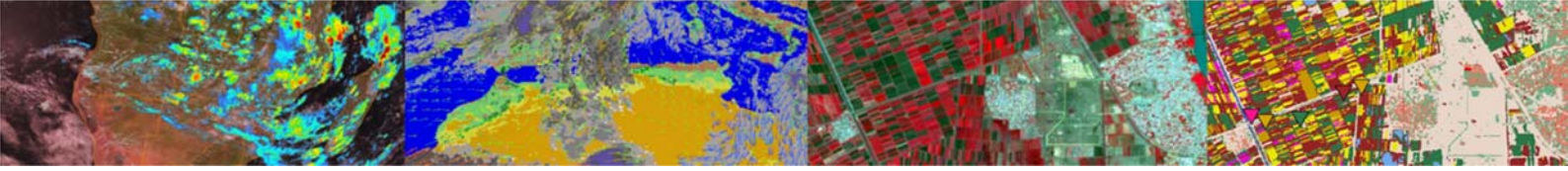


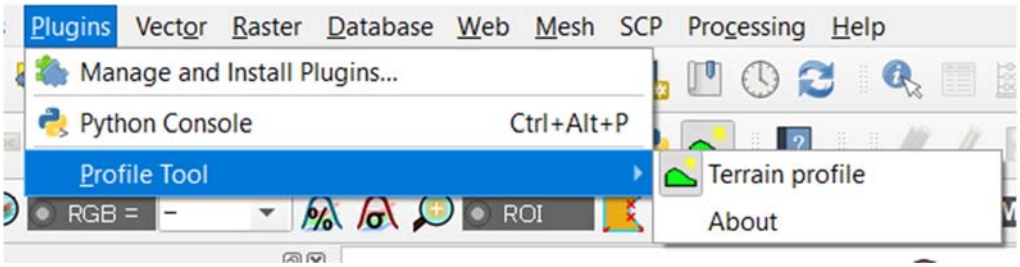
These images can be used directly without needing to perform atmospheric correction and reflectance conversion. First, the reflectance values in the Sentinel image are projected to the suspended sediment concentrations using the above regression equation by using Raster Calculator in QGIS. The product is a map of sediment concentration which is then used to analyze the sediment dynamics spatially and temporally.



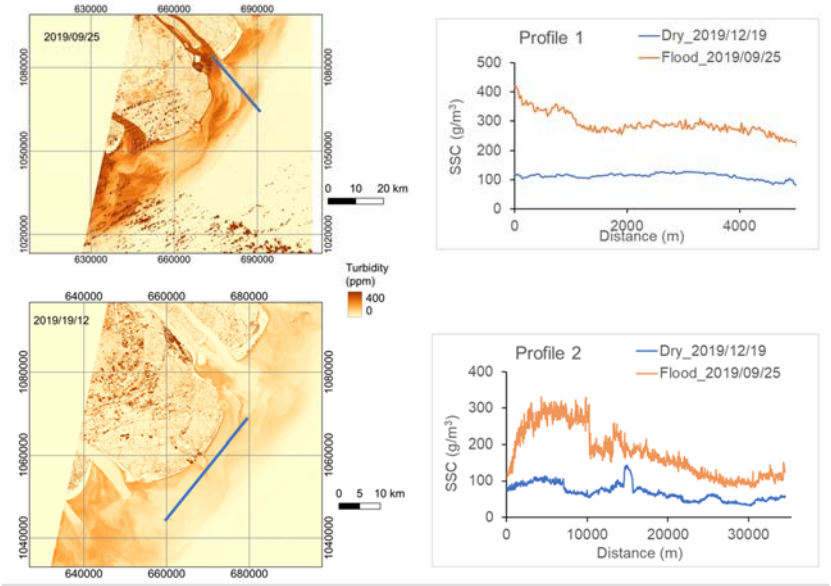
To extract some line profiles (longitudinal distributions) of the sediment concentration, the Profile Tool Plugin in QGIS is used.







Using this tool, we can draw some lines where we want to compare the profiles of different images. The results can be seen in the figure below



From this figure, we can see that suspended sediment concentrations in the flood season is much higher than those in the dry season. Therefore, sediment concentrations derived from Sentinel 2 is more reliable than those derived from Sentinel 3 without performing regression.

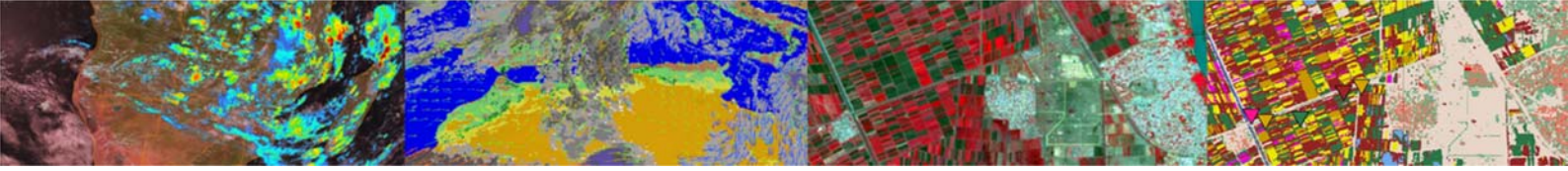
One limitation from the analysis is that the pixels covered by cloud provide abnormally high values of sediment. Therefore, the actual analysis must mask out the cloud pixels before conducting the analysis following the above-described steps.

Conclusion

Through this training course, I am able to used Sentinel to derive maps of suspended sediment concentration which is useful for my research in the future. More detailed and careful analyses will be performed in the actual study for journal paper publications.

Acknowledgement

I am sincerely thankful to Prof. Ben Maathuis for allocating time to train us and Juliette Eulderink for organizing the course.



3. Flood detection for Thua Thien-Hue province in October 2020 based on Sentinel-1 satellite imagery

Nghiem Tien Lam, Email: lamnt@wru.vn

Introduction

Thua Thien-Hue (TTH) province is located in the central coastal region of Vietnam. It is renowned for beautiful landscape, pristine beaches, and five UNESCO-recognized heritages including the complex of ancient palaces, ancient tombs, temples and shrines. The province is characterized as highly steeping topography with a coastal lowland and tidal lagoon, harsh climate and typhoon prone. Typhoons accompanied by torrential rainfall frequently flood the lowland areas which causes serious losses in human lives and properties, affect the social and economic development in the area.

In October and November 2020, a series of tropical cyclones such as Linfa, Nangka, Ofel, Saudel, and Molave, struck the area causing 49 deaths, many thousands houses flooded, and severe damages to crops, livestock, and infrastructure in the province. The total economic loss is estimated more than 100 million US dollars.

The information on the historical floods plays an important role to mitigate the damages and losses due to flooding in the area. This exercise is aimed to be acquainted with the application of remote sensing for flooding detection base on the Sentinel products.

Objective

The main objective of the work is to detect the flooding area using Sentinel satellite imagery. Specifically:

- To be familiar with the satellite datasets, data sources and data retrieval of the Sentinel products.
- To be familiar with the software and tools for Sentinel imagery processing.
- To be able to process the images to get the required information.

Research question

The research question is how to detect the flooding extent for a certain area based on Sentinel-1 satellite imagery?

Method and software

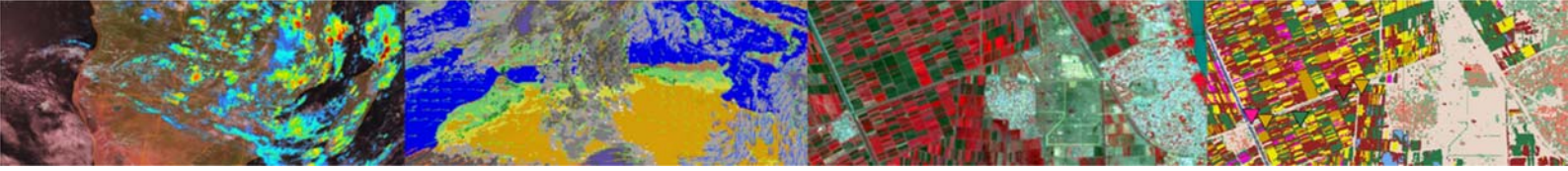
Based on the information provided in the course, the synthetic aperture radar (SAR) images from Sentinel-1 satellites are used to detect the flooding extent for the study area. Also JavaScript sample code was provided to use with Google Earth Engine (GEE).

The GEE JavaScript code runs well after having modified it for the selected study area. However, initially there is a small problem that the images exported from GEE cannot be found anywhere in neither Google Drive nor the Git repository. To overcome this the JavaScript code is converted to Jupyter Notebook to export the images to the local drive. Both GEE JavaScript (now updated for download) and Jupyter Notebook codes are listed in the Annexes.

Data Processing and Results

To determine the flood extent for the TTH area during the October 2020 flood, we use the 10 m resolution GRD data product processed at level 1 of the Sentinel-1A satellite in IW mode. The satellite data used is from the VH polarization channel during the descending pass.

Two image sets of before and after flood are filtered from September 23 - October 5, 2020 and October 6 - October 23, 2020, respectively. The entire process of selecting image data



according to the criteria of time, space, shooting mode, shooting channel, orbital direction, and resolution is carried out automatically from the available data set within GEE. The result is 8 satellite images with 3 images before the flood and 5 images during/after the flood. The times and locations of the images are as listed in Table 1 and are shown in Figure 1.

Table 1: Start time of the selected images

No.	Start time	Notes
1	2020-09-23 22:36:04	before flood
2	2020-09-28 22:43:41	before flood
3	2020-09-28 22:44:10	before flood
4	2020-10-10 22:43:41	after flood
5	2020-10-10 22:44:10	after flood
6	2020-10-17 22:36:04	after flood
7	2020-10-22 22:43:41	after flood
8	2020-10-22 22:44:10	after flood

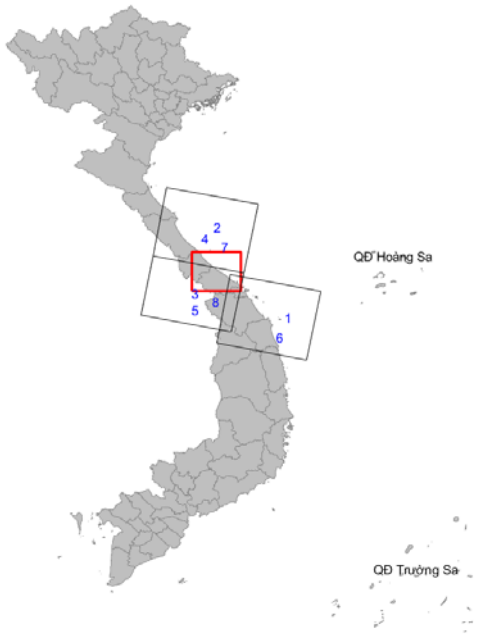


Figure 1: Footprint of the selected images

For each set of selected images, the images were merged and cropped for the area of interest into 2 images for the before and after flood periods. These two images were then smoothed using a morphological filter with a radius of 50 m to remove the radar speckles.

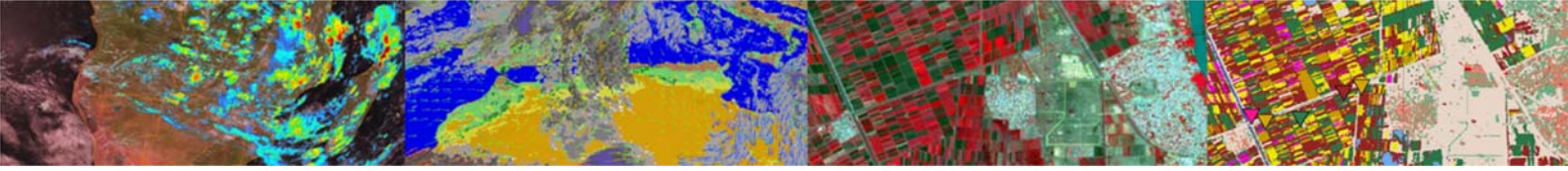
The flooded extent is determined by dividing the pixel values of those after the flood by the ones before the flood and keeping the pixel values greater than the threshold 1.2.

To determine the extent of inundation caused by the flood, the permanent water bodies such as lagoons, rivers, and lakes needed to be mask out. The permanent water bodies are identified as the areas with at least 10 months covered by water annually. This is determined based on a global water surface dataset produced by the Joint Research Center (JRC) of the European Commission based on Landsat satellite imagery over the past 37 years [3].

The next step to determine the extent of inundation is to superimpose the results obtained in the previous step with the digital elevation model [4] to eliminate areas with elevations of 10 m or higher and slopes of 5% or steeper. This is followed by a filtering step to remove flooded areas with less than or equal to 8 pixels.

Finally, the flood extent is converted from raster to vector in shapefile format.

With the use of GEE, it is also possible to conduct inundation impact assessment by superimposing with layers of population maps and land use maps. Figure 2 presents the results of the flood impact assessment when combined with the overlay dataset from the MODIS satellite images [2] and the 2015 population dataset of JRC [1] using GEE.



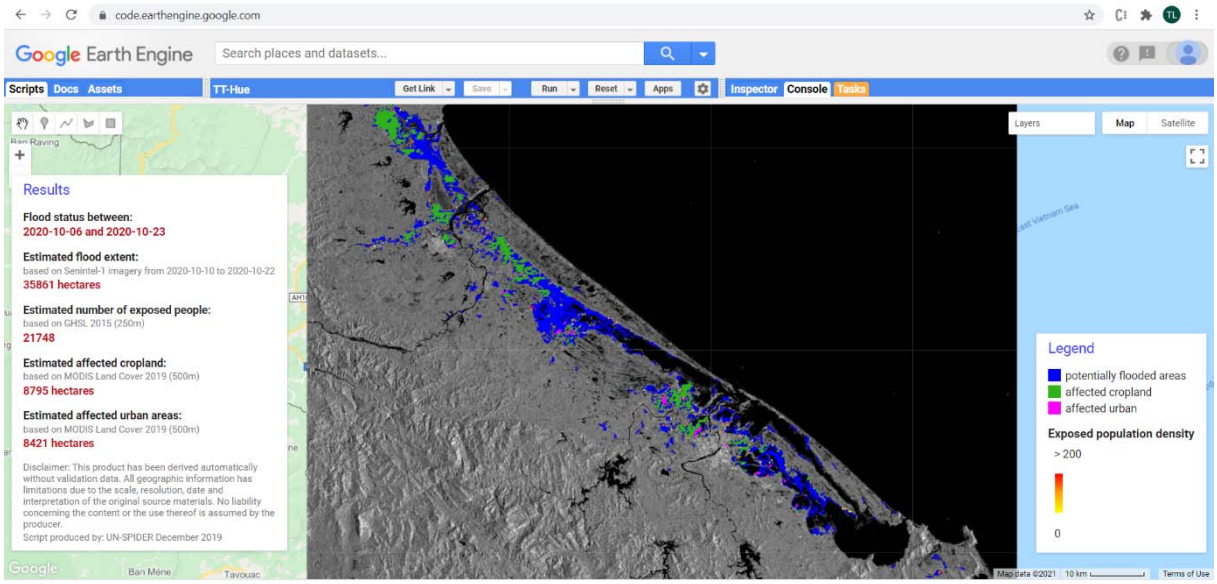


Figure 2: The flood extent determined using GEE.

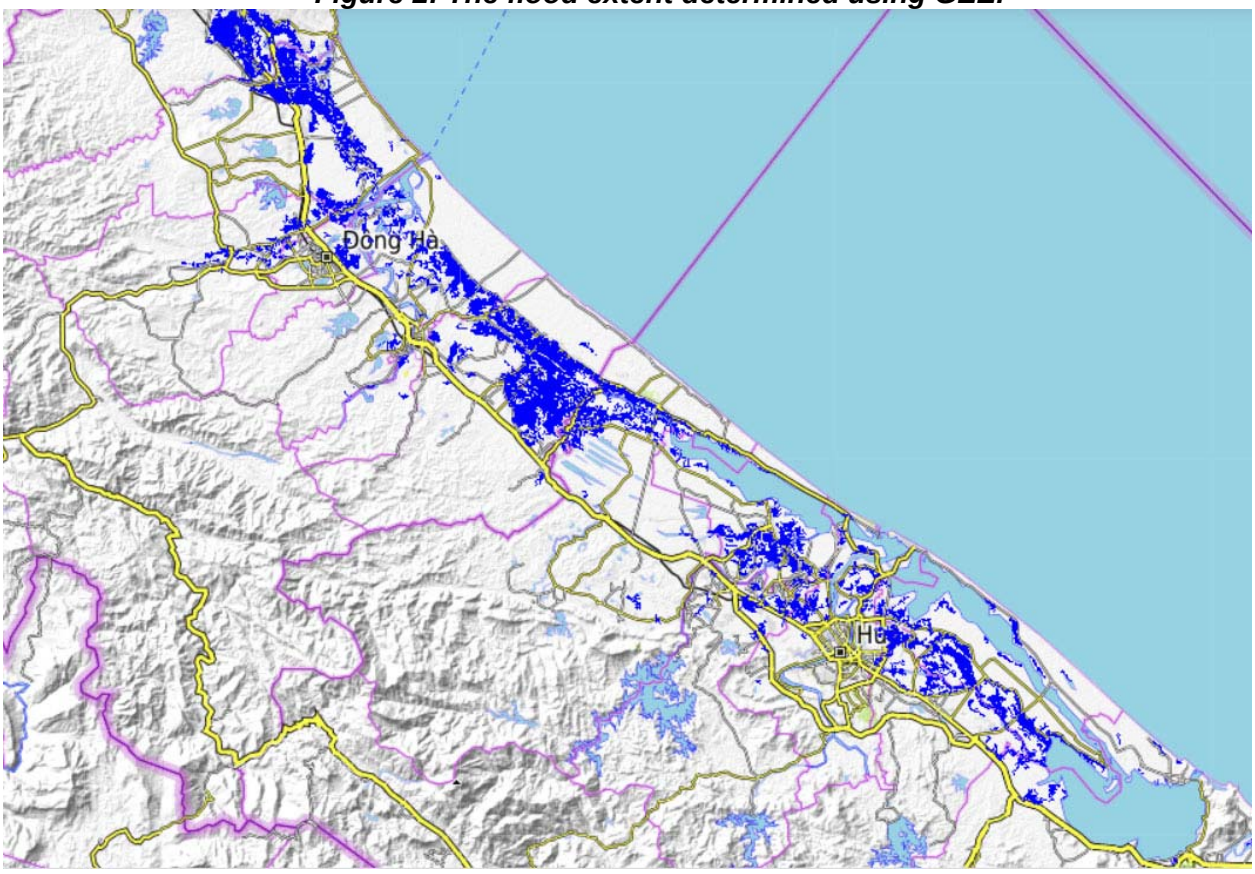


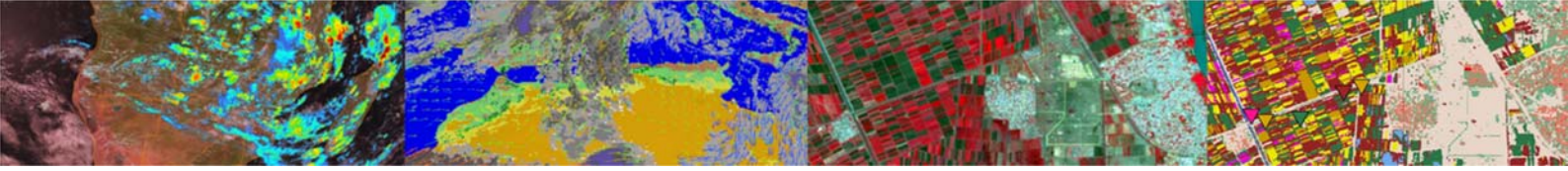
Figure 3: The flood extent determined using GEE in Jupyter Notebook.

Concluding Remarks

This report presents a small case study on the application of Sentinel satellite products to detect flood extent for Thua Thien-Hue province in Vietnam. This application is done with radar images from Sentinel 1 satellite and cloud computing using Google Earth Engine.

Through this small case study, I have learned:

- How to use GEE and Jupyter Notebook for processing datasets with could computing.
- How to retrieve and filter the image based on different criteria.



- How to process the data to get the flood extent from Sentinel imagery.

This small case study is conducted within the Nuffic sponsored 5 day training course “Use of ESA Sentinel satellite observations” for Vietnamese participants. The course provides the basic theoretical and hands-on parts on remote sensing, the Copernicus/Sentinel products, tools and data processing.

The application of cloud computing services through GEE services will exploit many large datasets and help determine the extent of flooding quickly, reducing the cost of retrieving, storing and processing the data.

This report cannot be done without the active guidance and support from the instructor, Prof. Ben Maathuis from the University of Twente and the coordinator, Ms. Juliette Eulderink.

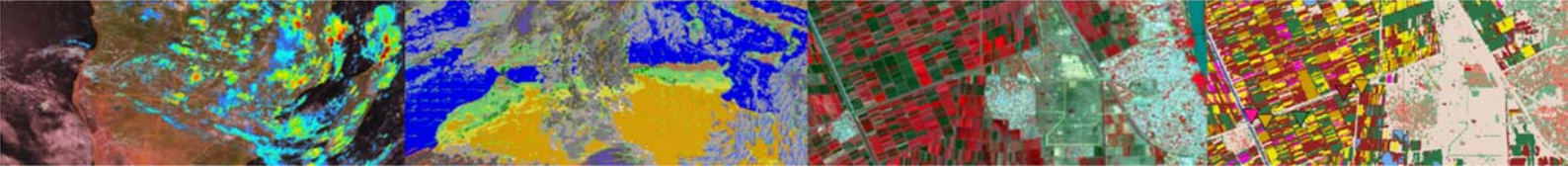
References

[1] European Commission, Joint Research Centre (JRC); Columbia University, Center for International Earth Science Information Network - CIESIN (2015): GHS population grid, derived from GPW4, multitemporal (1975, 1990, 2000, 2015). European Commission, Joint Research Centre (JRC)

[2] Friedl, M., Sulla-Menashe, D. (2019). MCD12Q1 MODIS/Terra+Aqua Land Cover Type Yearly L3 Global 500m SIN Grid V006 [Data set]. NASA EOSDIS Land Processes DAAC.

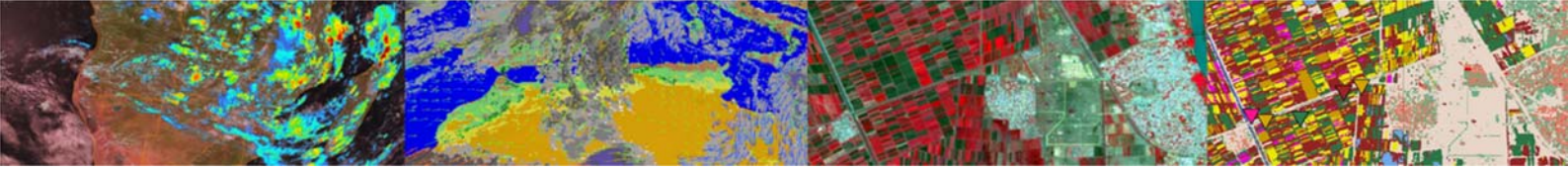
[3] Jean-Francois Pekel, Andrew Cottam, Noel Gorelick, Alan S. Belward, High-resolution mapping of global surface water and its long-term changes. Nature 540, 418-422 (2016).

[4] Lehner, B., Verdin, K., Jarvis, A. (2008): New global hydrography derived from spaceborne elevation data. Eos, Transactions, AGU, 89(10): 93-94.

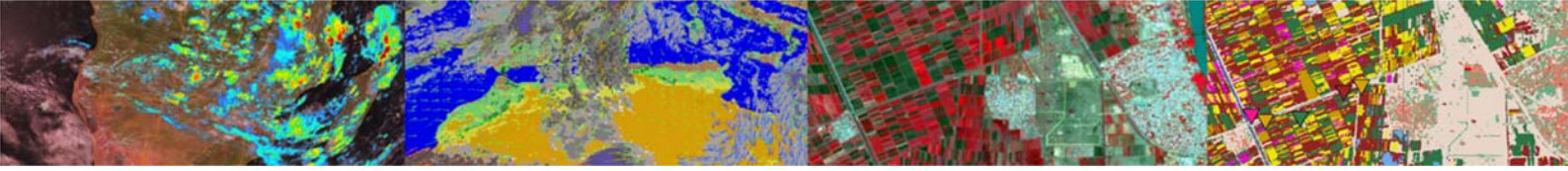


Annex1: JAVA GEE Script Code

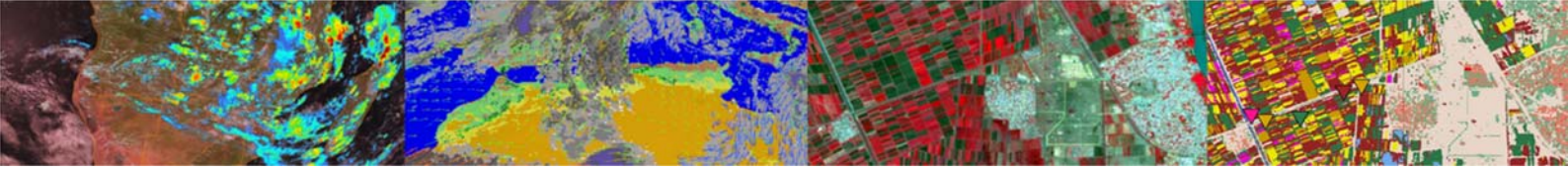
```
/**************************************************************************************************
                    SET REGION PARAMETERS*/

var geometry = ee.Geometry.Polygon([[
    [106.87471575068975,16.191956243532484],
    [108.11067766475225,16.191956243532484],
    [108.11067766475225,17.07617846943049],
    [106.87471575068975,17.07617846943049],
    [106.87471575068975,16.191956243532484]]]);

var before_start= '2020-09-23';
var before_end='2020-10-05';

// Now set the same parameters for AFTER the flood.
var after_start='2020-10-06';
var after_end='2020-10-23';

/**************************************************************************************************
                    SET SAR PARAMETERS*/

var polarization = "VH"; /*or 'VV' --> VH mostly is the preferred polarization for flood mapping.
                    However, it always depends on your study area, you can select 'VV'
                    as well.*/
var pass_direction = "DESCENDING"; /* or 'ASCENDING' when images are being compared use only
                   one pass direction. Consider changing this parameter, if your image
                    collection is empty. In some areas more Ascending images exist than
                    than descending or the other way around.*/
var difference_threshold = 1.2; /*1.25 threshold to be applied on the difference image (after flood
                    - before flood). It has been chosen by trial and error. In case your
                    flood extent result shows many false-positive or negative signals,
                    consider changing it! */
//var relative_orbit = 79;
                   /*if you know the relative orbit for your study area, you can filter
                    you image collection by it here, to avoid errors caused by comparing
                    different relative orbits.*/

//------------------------------ DATA SELECTION & PREPROCESSING -------------------------//

// rename selected geometry feature
var aoi = ee.FeatureCollection(geometry);

// Load and filter Sentinel-1 GRD data by predefined parameters
var collection= ee.ImageCollection('COPERNICUS/S1_GRD')
  .filter(ee.Filter.eq('instrumentMode','IW'))
  .filter(ee.Filter.listContains('transmitterReceiverPolarisation', polarization))
  .filter(ee.Filter.eq('orbitProperties_pass',pass_direction))
  .filter(ee.Filter.eq('resolution_meters',10))
  //.filter(ee.Filter.eq('relativeOrbitNumber_start',relative_orbit ))
  .filterBounds(aoi)
  .select(polarization);

// Select images by predefined dates
var before_collection = collection.filterDate(before_start, before_end);
var after_collection = collection.filterDate(after_start,after_end);

// Print selected tiles to the console

// Extract date from meta data
function dates(imgcol){
```



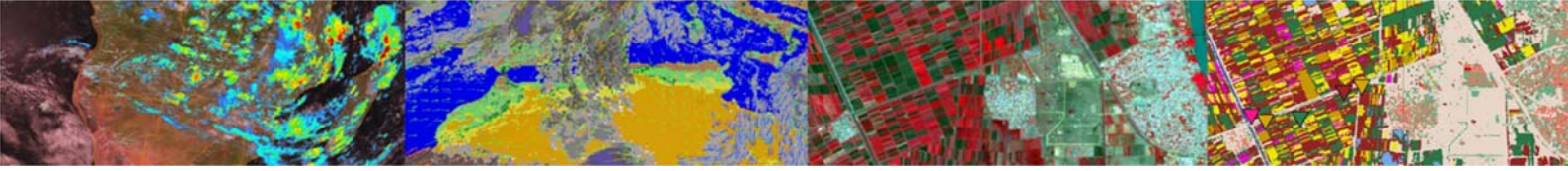
```
  var range = imgcol.reduceColumns(ee.Reducer.minMax(), ["system:time_start"]);
  var printed = ee.String('from ').cat(ee.Date(range.get('min')).format('YYYY-MM-dd'))
    .cat(' to ').cat(ee.Date(range.get('max')).format('YYYY-MM-dd'));
  return printed;
}
// print dates of before images to console
var before_count = before_collection.size();
print(ee.String('Tiles selected: Before Flood ').cat('(').cat(before_count).cat(')'),
  dates(before_collection), before_collection);

// print dates of after images to console
var after_count = before_collection.size();
print(ee.String('Tiles selected: After Flood ').cat('(').cat(after_count).cat(')'),
  dates(after_collection), after_collection);

// Create a mosaic of selected tiles and clip to study area
var before = before_collection.mosaic().clip(aoi);
var after = after_collection.mosaic().clip(aoi);

// Apply reduce the radar speckle by smoothing
var smoothing_radius = 50;
var before_filtered = before.focal_mean(smoothing_radius, 'circle', 'meters');
var after_filtered = after.focal_mean(smoothing_radius, 'circle', 'meters');


//------------------------------ FLOOD EXTENT CALCULATION ------------------------------//

// Calculate the difference between the before and after images
var difference = after_filtered.divide(before_filtered);

// Apply the predefined difference-threshold and create the flood extent mask
var threshold = difference_threshold;
var difference_binary = difference.gt(threshold);

// Refine flood result using additional datasets

// Include JRC layer on surface water seasonality to mask flood pixels from areas
// of "permanent" water (where there is water > 10 months of the year)
var swater = ee.Image('JRC/GSW1_0/GlobalSurfaceWater').select('seasonality');
var swater_mask = swater.gte(10).updateMask(swater.gte(10));

//Flooded layer where perennial water bodies (water > 10 mo/yr) is assigned a 0 value
var flooded_mask = difference_binary.where(swater_mask,0);
// final flooded area without pixels in perennial waterbodies
var flooded = flooded_mask.updateMask(flooded_mask);

// Compute connectivity of pixels to eliminate those connected to 8 or fewer neighbours
// This operation reduces noise of the flood extent product
var connections = flooded.connectedPixelCount();
var flooded = flooded.updateMask(connections.gte(8));

// Mask out areas with more than 5 percent slope using a Digital Elevation Model
var DEM = ee.Image('WWF/HydroSHEDS/03VFDEM');
var terrain = ee.Algorithms.Terrain(DEM);
var elevation = terrain.select('b1');
var slope = terrain.select('slope');
var flooded = flooded.updateMask(slope.lt(5) && elevation.lt(10));

// Calculate flood extent area
// Create a raster layer containing the area information of each pixel
var flood_pixelarea = flooded.select(polarization)
```



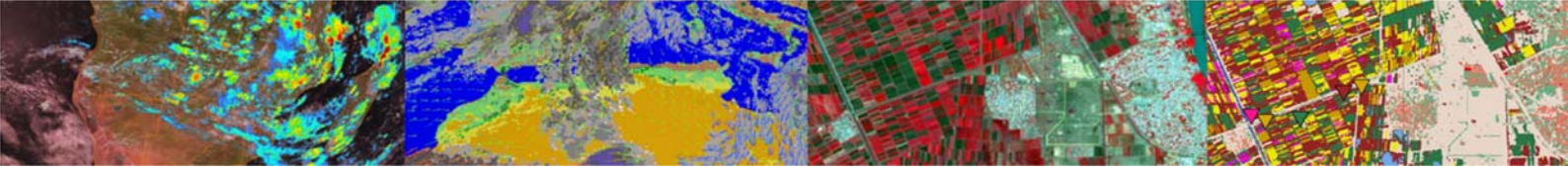
```
  .multiply(ee.Image.pixelArea());

// Sum the areas of flooded pixels
// default is set to 'bestEffort: true' in order to reduce compuation time, for a more
// accurate result set bestEffort to false and increase 'maxPixels'.
var flood_stats = flood_pixelarea.reduceRegion({
  reducer: ee.Reducer.sum(),
  geometry: aoi,
  scale: 10, // native resolution
  //maxPixels: 1e9,
  bestEffort: true
  });

// Convert the flood extent to hectares (area calculations are originally given in meters)
var flood_area_ha = flood_stats.getNumber(polarization).divide(10000).round();


//------------------------------  DAMAGE ASSSESSMENT  ---------------------------------//

//------------------------------ Exposed population density ----------------------------//

// Load JRC Global Human Settlement Popluation Density layer
// Resolution: 250. Number of people per cell is given.
var population_count = ee.Image('JRC/GHSL/P2016/POP_GPW_GLOBE_V1/2015').clip(aoi);

// Calculate the amount of exposed population
// get GHSL projection
var GHSLprojection = population_count.projection();

// Reproject flood layer to GHSL scale
var flooded_res1 = flooded.reproject({ crs: GHSLprojection });

// Create a raster showing exposed population only using the resampled flood layer
var population_exposed = population_count
  .updateMask(flooded_res1).updateMask(population_count);

//Sum pixel values of exposed population raster
var stats = population_exposed.reduceRegion({
  reducer: ee.Reducer.sum(),
  geometry: aoi,
  scale: 250,
  maxPixels:1e9
});

// get number of exposed people as integer
var number_pp_exposed = stats.getNumber('population_count').round();

//------------------------------ Affected agricultural land ----------------------------//

// using MODIS Land Cover Type Yearly Global 500m
// filter image collection by the most up-to-date MODIS Land Cover product
var LC = ee.ImageCollection('MODIS/006/MCD12Q1')
  .filterDate('2014-01-01',after_end).sort('system:index',false)
  .select("LC_Type1").first().clip(aoi);

// Extract only cropland pixels using the classes cropland (>60%) and Cropland/Natural
// Vegetation Mosaics: mosaics of small-scale cultivation 40-60% incl. natural vegetation
var cropmask = LC.eq(12).or(LC.eq(14))
var cropland = LC.updateMask(cropmask)

// get MODIS projection
```



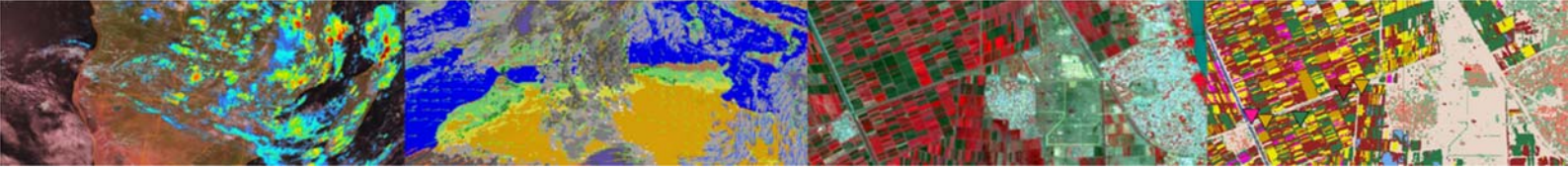
```
var MODISprojection = LC.projection();

// Reproject flood layer to MODIS scale
var flooded_res = flooded.reproject({crs: MODISprojection });

// Calculate affected cropland using the resampled flood layer
var cropland_affected = flooded_res.updateMask(cropland)

// get pixel area of affected cropland layer
var crop_pixelarea = cropland_affected
  .multiply(ee.Image.pixelArea()); //calcuate the area of each pixel

// sum pixels of affected cropland layer
var crop_stats = crop_pixelarea.reduceRegion({
  reducer: ee.Reducer.sum(), //sum all pixels with area information
  geometry: aoi,
  scale: 500,
  maxPixels: 1e9
  });

// convert area to hectares
var crop_area_ha = crop_stats.getNumber(polarization).divide(10000).round();

//------------------------------- Affected urban area ------------------------------//

// Using the same MODIS Land Cover Product
// Filter urban areas
var urbanmask = LC.eq(13)
var urban = LC.updateMask(urbanmask)

//Calculate affected urban areas using the resampled flood layer
var urban_affected = urban.mask(flooded_res).updateMask(urban);

// get pixel area of affected urban layer
var urban_pixelarea = urban_affected
  .multiply(ee.Image.pixelArea()); //calcuate the area of each pixel

// sum pixels of affected cropland layer
var urban_stats = urban_pixelarea.reduceRegion({
  reducer: ee.Reducer.sum(), //sum all pixels with area information
  geometry: aoi,
  scale: 500,
  bestEffort: true,
  });

// convert area to hectares
var urban_area_ha = urban_stats.getNumber('LC_Type1').divide(10000).round();

//------------------------------  DISPLAY PRODUCTS  ---------------------------------//

// Before and after flood SAR mosaic
Map.centerObject(aoi,8);
Map.addLayer(before_filtered, {min:-25,max:0}, 'Before Flood',0);
Map.addLayer(after_filtered, {min:-25,max:0}, 'After Flood',1);

// Difference layer
Map.addLayer(difference,{min:0,max:2},"Difference Layer",0);

// Flooded areas
Map.addLayer(flooded,{palette:"0000FF"},'Flooded areas');
```



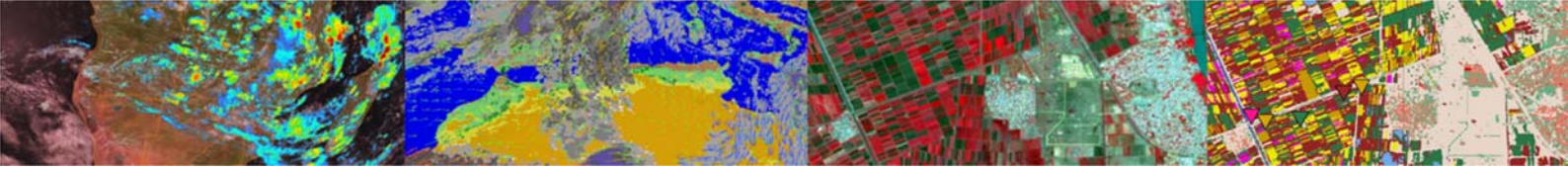
```
// Population Density
var populationCountVis = { min: 0, max: 200.0,
  palette: ['060606','337663','337663','ffffff'],
};
Map.addLayer(population_count, populationCountVis, 'Population Density',0);

// Exposed Population
var populationExposedVis = { min: 0, max: 200.0,
  palette: ['yellow', 'orange', 'red'],
};
Map.addLayer(population_exposed, populationExposedVis, 'Exposed Population');

// MODIS Land Cover
var LCVis = { min: 1.0, max: 17.0,
  palette: [
    '05450a', '086a10', '54a708', '78d203', '009900', 'c6b044', 'dcd159',
    'dade48', 'fbff13', 'b6ff05', '27ff87', 'c24f44', 'a5a5a5', 'ff6d4c',
    '69fff8', 'f9ffa4', '1c0dff'
  ],
};
Map.addLayer(LC, LCVis, 'Land Cover',0);

// Cropland
var croplandVis = { min: 0, max: 14.0, palette: ['30b21c'],};
Map.addLayer(cropland, croplandVis, 'Cropland',0)

// Affected cropland
Map.addLayer(cropland_affected, croplandVis, 'Affected Cropland');

// Urban
var urbanVis = {min: 0, max: 13.0, palette: ['magenta'], };
Map.addLayer(urban, urbanVis, 'Urban',0)

// Affected urban
Map.addLayer(urban_affected, urbanVis, 'Affected Urban');


//------------------------------------ EXPORTS ------------------------------------//

// Export flooded area as TIFF file
Export.image.toDrive({
  image: flooded,
  description: 'Flood_extent',
  region: aoi,
  scale: 100,
  maxPixels: 2000000000
});

// Export flooded area as shapefile (for further analysis in e.g. QGIS)
// Convert flood raster to polygons
var flooded_vec = flooded.reduceToVectors({
  scale: 10,
  geometryType:'polygon',
  geometry: aoi,
  eightConnected: false,
  bestEffort:true,
  tileScale:2,
});

// Export flood polygons as shape-file
Export.table.toDrive({
```



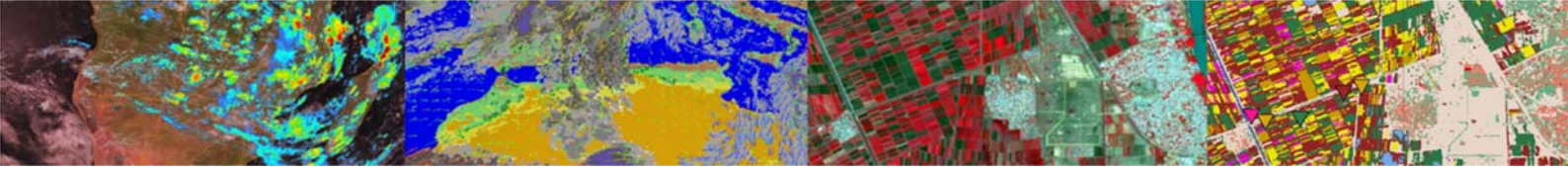
```
  collection:flooded_vec,
  description:'Flood_extent_vector',
  fileFormat:'SHP',
  fileNamePrefix:'flooded_vec'
});

// Export auxcillary data as shp?
// Exposed population density
Export.image.toDrive({
  image:population_exposed,
  description:'Exposed_Populuation',
  scale: 250,
  fileNamePrefix:'population_exposed',
  region: aoi,
  maxPixels:1e10
});

//---------------------------------- MAP PRODUCTION ---------------------------------//

//-------------------------- Display the results on the map ----------------------//

// set position of panel where the results will be displayed
var results = ui.Panel({
  style: { position: 'bottom-left', padding: '8px 15px', width: '350px' }
});

//Prepare the visualtization parameters of the labels
var textVis = { 'margin':'0px 8px 2px 0px', 'fontWeight':'bold' };
var numberVIS = { 'margin':'0px 0px 15px 0px',  'color':'bf0f19', 'fontWeight':'bold' };
var subTextVis = { 'margin':'0px 0px 2px 0px', 'fontSize':'12px', 'color':'grey' };

var titleTextVis = { 'margin':'0px 0px 15px 0px', 'fontSize': '18px',
  'font-weight':'',  'color': '3333ff' };

// Create lables of the results
// Titel and time period
var title = ui.Label('Results', titleTextVis);
var text1 = ui.Label('Flood status between:',textVis);
var number1 = ui.Label(after_start.concat(" and ",after_end),numberVIS);

// Alternatively, print dates of the selected tiles
//var number1 = ui.Label('Please wait...',numberVIS);
//(after_collection).evaluate(function(val){number1.setValue(val)}),numberVIS;

// Estimated flood extent
var text2 = ui.Label('Estimated flood extent:',textVis);
var text2_2 = ui.Label('Please wait...',subTextVis);
dates(after_collection).evaluate(function(val){text2_2.setValue('based on Senintel-1 imagery '+val)});
var number2 = ui.Label('Please wait...',numberVIS);
flood_area_ha.evaluate(function(val){number2.setValue(val+' hectares')}),numberVIS;

// Estimated number of exposed people
var text3 = ui.Label('Estimated number of exposed people: ',textVis);
var text3_2 = ui.Label('based on GHSL 2015 (250m)',subTextVis);
var number3 = ui.Label('Please wait...',numberVIS);
number_pp_exposed.evaluate(function(val){number3.setValue(val)}),numberVIS;

// Estimated area of affected cropland
var MODIS_date = ee.String(LC.get('system:index')).slice(0,4);
var text4 = ui.Label('Estimated affected cropland:',textVis);
var text4_2 = ui.Label('Please wait', subTextVis)
```



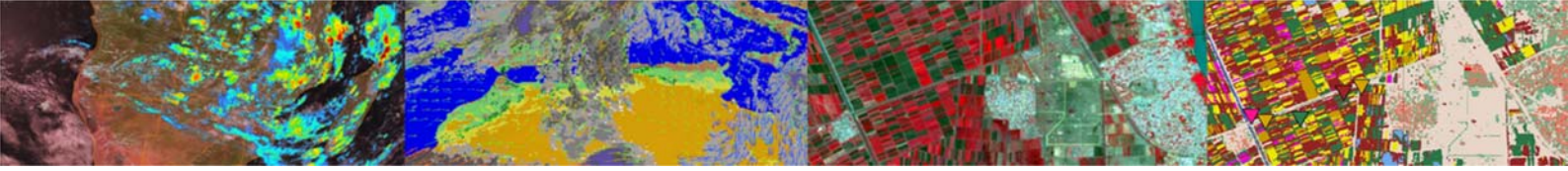
```
MODIS_date.evaluate(function(val){text4_2.setValue('based on MODIS Land Cover '+val +' (500m)')}),
subTextVis;
var number4 = ui.Label('Please wait...',numberVIS);
crop_area_ha.evaluate(function(val){number4.setValue(val+' hectares')}),numberVIS;

// Estimated area of affected urban
var text5 = ui.Label('Estimated affected urban areas:',textVis);
var text5_2 = ui.Label('Please wait', subTextVis)
MODIS_date.evaluate(function(val){text5_2.setValue('based on MODIS Land Cover '+val +' (500m)')}),
subTextVis;
var number5 = ui.Label('Please wait...',numberVIS);
urban_area_ha.evaluate(function(val){number5.setValue(val+' hectares')}),numberVIS;

// Disclaimer
var text6 = ui.Label('Disclaimer: This product has been derived automatically without validation data. All
geographic information has limitations due to the scale, resolution, date and interpretation of the original
source materials. No liability concerning the content or the use thereof is assumed by the
producer.',subTextVis)

// Produced by...
var text7 = ui.Label('Script produced by: UN-SPIDER December 2019', subTextVis)

// Add the labels to the panel
results.add(ui.Panel([
      title,
      text1, number1,
      text2, text2_2,
      number2,
      text3,
      text3_2,
      number3,
      text4,
      text4_2,
      number4,
      text5,
      text5_2,
      number5,
      text6,
      text7]
    ));

// Add the panel to the map
Map.add(results);

//----------------------------- Display legend on the map -------------------------//

// Create legend (*credits to thisearthsite on Open Geo Blog: https://mygeoblog.com/2016/12/09/add-a-
legend-to-to-your-gee-map/)
// set position of panel
var legend = ui.Panel({
  style: { position: 'bottom-right', padding: '8px 15px', }
});

// Create legend title
var legendTitle = ui.Label('Legend',titleTextVis);

// Add the title to the panel
legend.add(legendTitle);

// Creates and styles 1 row of the legend.
var makeRow = function(color, name) {
```



```
    // Create the label that is actually the colored box.
    var colorBox = ui.Label({
      style: {
        backgroundColor: color,
        // Use padding to give the box height and width.
        padding: '8px',
        margin: '0 0 4px 0'
      }
    });

    // Create the label filled with the description text.
    var description = ui.Label({
      value: name,
      style: {margin: '0 0 4px 6px'}
    });

    // return the panel
    return ui.Panel({
      widgets: [colorBox, description],
      layout: ui.Panel.Layout.Flow('horizontal')
    });
};

//  Palette with the colors
var palette =['#0000FF', '#30b21c', 'magenta'];

// name of the legend
var names = ['potentially flooded areas','affected cropland','affected urban'];

// Add color and and names
for (var i = 0; i < 3; i++) {
  legend.add(makeRow(palette[i], names[i]));
  }

// Create second legend title to display exposed population density
var legendTitle2 = ui.Label({
value: 'Exposed population density',
style: {
fontWeight: 'bold',
fontSize: '15px',
margin: '10px 0 0 0',
padding: '0'
}
});

// Add second title to the panel
legend.add(legendTitle2);

// create the legend image
var lon = ee.Image.pixelLonLat().select('latitude');
var gradient = lon.multiply((populationExposedVis.max-
populationExposedVis.min)/100.0).add(populationExposedVis.min);
var legendImage = gradient.visualize(populationExposedVis);

// create text on top of legend
var panel = ui.Panel({
widgets: [
ui.Label('> '.concat(populationExposedVis['max']))
],
});
```



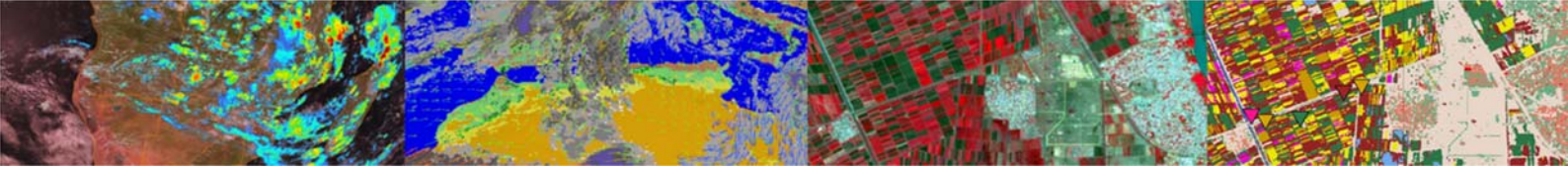
```
legend.add(panel);

// create thumbnail from the image
var thumbnail = ui.Thumbnail({
image: legendImage,
params: {bbox:'0,0,10,100', dimensions:'10x50'},
style: {padding: '1px', position: 'bottom-center'}
});

// add the thumbnail to the legend
legend.add(thumbnail);

// create text on top of legend
var panel = ui.Panel({
widgets: [
ui.Label(populationExposedVis['min'])
],
});

legend.add(panel);

// add legend to map (alternatively you can also print the legend to the console)
Map.add(legend);
```



Annex 2: Jupyter Notebook code asci listing

== Flood Mapping for Thua Thien-Hue Province==

```
---
import geemap
import ee
#ee.Authenticate()
ee.Initialize()
---

---
roi = ee.Geometry.Rectangle(106.875, 16.10, 108.125, 17.10)
# The period before the flood
before_start= ee.Date('2020-09-23');
before_end=ee.Date('2020-10-05');

# The period after the flood
after_start=ee.Date('2020-10-06');
after_end=ee.Date('2020-10-23');
---

---
polarization = "VH";

# Import the Sentinel 1 collection
s1 = ee.ImageCollection("COPERNICUS/S1_GRD").filterBounds(roi) \
      .filter(ee.Filter.eq('instrumentMode','IW')) \
      .filter(ee.Filter.eq('orbitProperties_pass', "DESCENDING")) \
      .filter(ee.Filter.eq('resolution_meters',10)) \

ic = s1.select(polarization) \
    .filter(ee.Filter.listContains('transmitterReceiverPolarisation', polarization))

# Filter the image collection usingfilterDate() method.
# Select images by predefined dates
before_collection = ic.filterDate(before_start, before_end);
after_collection = ic.filterDate(after_start,after_end);

# Create the mosaic image clip it to the ROI
img_before = before_collection.mosaic().clip(roi)
img_after = after_collection.mosaic().clip(roi)

#
exp_res = 200
exp_crs = "EPSG:4326"
#geemap.ee_export_image(img_before, '01-before.tif', exp_res, exp_crs, roi, 1);
#geemap.ee_export_image(img_after, '02-after.tif', exp_res, exp_crs, roi, 1);
---

---
# Apply reduce the radar speckle by smoothing
smoothing_radius = 50;
before_filtered = img_before.focal_mean(smoothing_radius, 'circle', 'meters');
after_filtered = img_after.focal_mean(smoothing_radius, 'circle', 'meters');
#geemap.ee_export_image(before_filtered, '03-before_filtered.tif', exp_res, exp_crs, roi, 1);
#geemap.ee_export_image(after_filtered, '04-after_filtered.tif', exp_res, exp_crs, roi, 1);
---

---
# Calculate the difference between the before and after images
```



```
difference = after_filtered.divide(before_filtered)

# Apply the predefined difference-threshold and create the flood extent mask
difference_threshold = 1.20;
img_diff = difference.gt(difference_threshold);
#geemap.ee_export_image(difference, '05-difference.tif', exp_res, exp_crs, roi, 1);
#geemap.ee_export_image(img_diff, '06-difference_threshold.tif', exp_res, exp_crs, roi, 1);
```

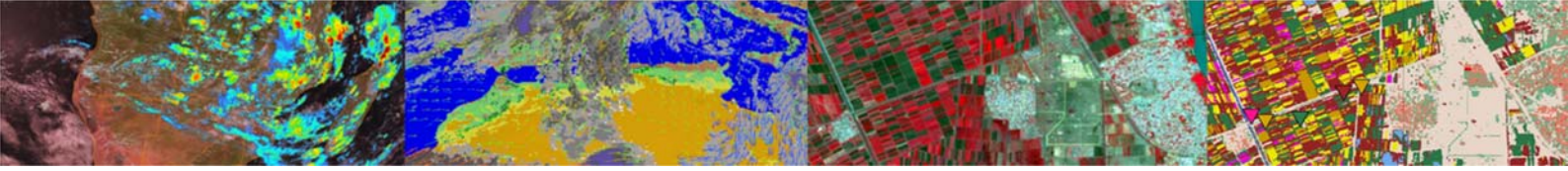
```
Map1 = geemap.Map(center = [16.6, 107.5], zoom = 11)
Map1.add_basemap('OpenTopoMap')
#Map1.add_basemap('HYBRID')
#Map1.addLayer(roi, {}, 'Area of Interest');
#Map1.addLayer(img_before , {}, 'Before the flood');
#Map1.addLayer(img_after , {}, 'After the flood');
Map1.addLayer(img_diff , {}, 'Difference');
Map1
```

== Refine flood result using additional datasets

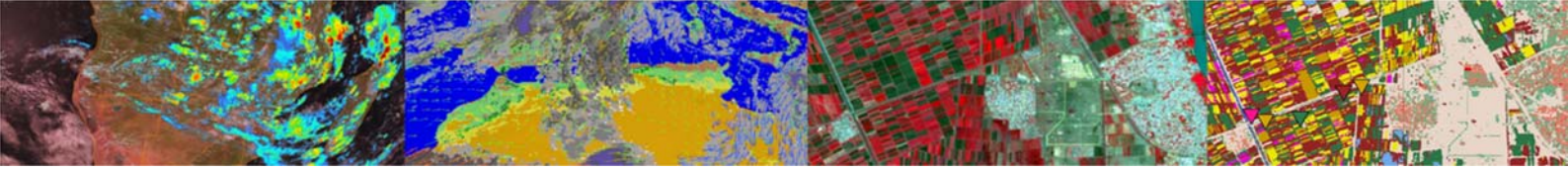
```
# Include JRC layer on surface water seasonality to mask flood pixels from areas
# of "permanent" water (where there is water > 10 months of the year)
swater = ee.Image('JRC/GSW1_0/GlobalSurfaceWater').select('seasonality');
swater_mask = swater.gte(10).updateMask(swater.gte(10));

# Flooded layer where perennial water bodies (water > 10 mo/yr) is assigned a 0 value
flooded_mask = img_diff.where(swater_mask,0);

#geemap.ee_export_image(swater_mask, '07-water_mask.tif', exp_res, exp_crs, roi, 1);
#geemap.ee_export_image(flooded_mask, '08-flooded_mask.tif', exp_res, exp_crs, roi, 1);

#final flooded area without pixels in perennial waterbodies
flooded = flooded_mask.updateMask(flooded_mask);

#geemap.ee_export_image(flooded, '09-flooded.tif', exp_res, exp_crs, roi, 1);

# Compute connectivity of pixels to eliminate those connected to 8 or fewer neighbours
# This operation reduces noise of the flood extent product
connections = flooded.connectedPixelCount();
flooded = flooded.updateMask(connections.gte(8));

#geemap.ee_export_image(flooded, '10-flooded-conn.tif', exp_res, exp_crs, roi, 1);
```

```
# Export flooded area as shapefile (for further analysis in e.g. QGIS)
# Convert flood raster to polygons
flooded_vec = flooded.reduceToVectors(scale = 10, geometryType = 'polygon', geometry = roi,
  eightConnected = False, bestEffort = True, tileScale = 2);
```

```
# Mask out areas with more than 5 percent slope using a Digital Elevation Model
DEM = ee.Image('WWF/HydroSHEDS/03VFDEM');
terrain = ee.Algorithms.Terrain(DEM);
elevation = terrain.select('b1');
slope = terrain.select('slope');
topo_mask = slope.lt(5).bitwiseAnd(elevation.lt(10));
flooded = flooded.updateMask(topo_mask);
```



```
#geemap.ee_export_image(topo_mask, '11-topo_mask.tif', exp_res, exp_crs, roi, 1);
#geemap.ee_export_image(flooded, '12-flooded.tif', exp_res, exp_crs, roi, 1);

Map2 = geemap.Map(center = [16.6, 107.5], zoom = 11)
Map2.add_basemap('OpenTopoMap')
#Map2.add_basemap('HYBRID')
#Map2.addLayer(roi, {}, 'Area of Interest');
#Map2.addLayer(img_before , {}, 'Before the flood');
#Map2.addLayer(img_after , {}, 'After the flood');
Map2.addLayer(flooded , { 'palette': ['blue'], min: 0.0, max: 1.0,}, 'Flooded area');
#Map2
----

== Calculate flood extent area==

----
# Create a raster layer containing the area information of each pixel
flood_pixelarea = flooded.select(polarization).multiply(ee.Image.pixelArea());

# Sum the areas of flooded pixels
# default is set to 'bestEffort: true' in order to reduce compuation time, for a more
# accurate result set bestEffort to false and increase 'maxPixels'.
flood_stats = flood_pixelarea.reduceRegion(
  reducer = ee.Reducer.sum(),
  geometry = roi,
  scale = 10,
  bestEffort = True);

# Convert the flood extent to hectares (area calculations are originally given in meters)
flood_area_ha = flood_stats.getNumber(polarization).divide(10000).round();

print('Estimated flood extent: ', flood_area_ha.getInfo())
----

== Damage assessment==

== Exposed population density==

----
# Load JRC Global Human Settlement Popluation Density layer
# Resolution: 250. Number of people per cell is given.
population_count = ee.Image('JRC/GHSL/P2016/POP_GPW_GLOBE_V1/2015').clip(roi);

# Calculate the amount of exposed population
# get GHSL projection
GHSLprojection = population_count.projection();

# Reproject flood layer to GHSL scale
flooded_res1 = flooded.reproject(crs = "EPSG:3857"); #GHSLprojection, "EPSG:54009"

# Create a raster showing exposed population only using the resampled flood layer
population_exposed = population_count.updateMask(flooded_res1).updateMask(population_count);

# Sum pixel values of exposed population raster
stats = population_exposed.reduceRegion(
    reducer = ee.Reducer.sum(),
    geometry = roi,  scale = 250, maxPixels = 1e9 );

# # get number of exposed people as integer
number_pp_exposed = stats.getNumber('population_count').round();
```



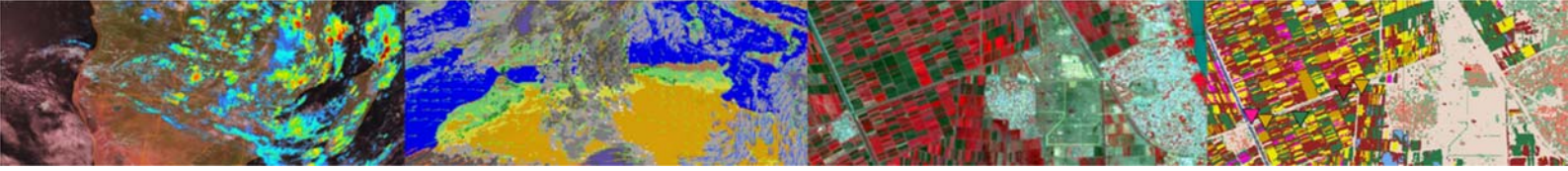
```
#print('Estimated number of exposed people: ', number_pp_exposed)
```

== Affected agricultural land==

```
# using MODIS Land Cover Type Yearly Global 500m
# filter image collection by the most up-to-date MODIS Land Cover product
LC = ee.ImageCollection('MODIS/006/MCD12Q1') \
     .filterDate('2014-01-01',after_end).sort('system:index', False) \
     .select("LC_Type1").first().clip(roi);

# Extract only cropland pixels using the classes cropland (>60%) and Cropland/Natural
# Vegetation Mosaics: mosaics of small-scale cultivation 40-60% incl. natural vegetation
cropmask = LC.eq(12).bitwiseOr(LC.eq(14))
cropland = LC.updateMask(cropmask)

# get MODIS projection
MODISprojection = LC.projection();

# Reproject flood layer to MODIS scale
flooded_res = flooded.reproject(crs = "EPSG:3857"); #MODISprojection //EPSG:6842, "EPSG:6974"

# Calculate affected cropland using the resampled flood layer
cropland_affected = flooded_res.updateMask(cropland)

# get pixel area of affected cropland layer
crop_pixelarea = cropland_affected.multiply(ee.Image.pixelArea()); # calculate the area of each pixel

# sum pixels of affected cropland layer
crop_stats = crop_pixelarea.reduceRegion(
   reducer = ee.Reducer.sum(), #sum all pixels with area information
   geometry = roi, scale = 500, maxPixels = 1e9 );

# convert area to hectares
crop_area_ha = crop_stats.getNumber(polarization).divide(10000).round();

print('Estimated affected cropland: ', crop_area_ha.getInfo())
```

== Affected urban area==

```
# Using the same MODIS Land Cover Product
# Filter urban areas
urbanmask = LC.eq(13)
urban = LC.updateMask(urbanmask)

# Calculate affected urban areas using the resampled flood layer
urban_affected = urban.mask(flooded_res).updateMask(urban);

# get pixel area of affected urban layer
urban_pixelarea = urban_affected.multiply(ee.Image.pixelArea()); # calcuate the area of each pixel

# sum pixels of affected cropland layer
urban_stats = urban_pixelarea.reduceRegion(
  reducer = ee.Reducer.sum(), # sum all pixels with area information
  geometry = roi, scale = 500, bestEffort = True);

# convert area to hectares
```



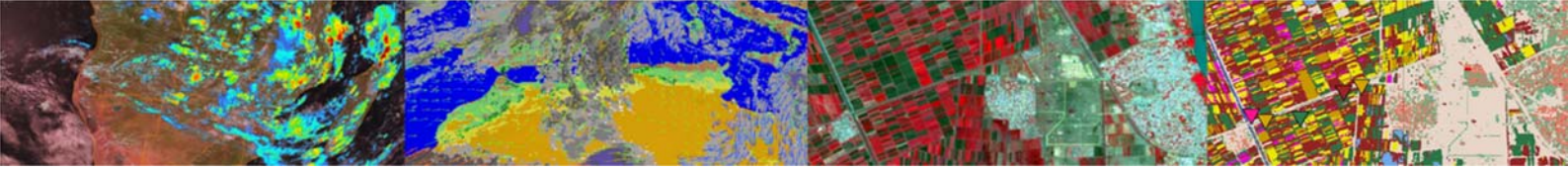
```
urban_area_ha = urban_stats.getNumber('LC_Type1').divide(10000).round();

#print('Estimated affected urban areas: ', urban_area_ha.getInfo())
```

== Display products==

```
Map = geemap.Map(center = [16.6, 107.5], zoom = 11)
Map.add_basemap('OpenTopoMap')
#Map.add_basemap('HYBRID')
#Map.addLayer(roi, {}, 'Area of Interest');
#Map.addLayer(img_before , {}, 'Before the flood');
#Map.addLayer(img_after , {}, 'After the flood');
#Map.addLayer(img_diff , {}, 'Difference');

# Before and after flood SAR mosaic
Map.centerObject(roi,8);
#Map.addLayer(before_filtered, {min:-25,max:0}, 'Before Flood',0);
#Map.addLayer(after_filtered, {min:-25,max:0}, 'After Flood',1);

# Difference layer
#Map.addLayer(img_diff, {min:0,max:2},"Difference Layer",0);

# Flooded areas
Map.addLayer(flooded,{'palette':"0000FF"},'Flooded areas');

# Population Density
Map.addLayer(population_count, { min: 0, max: 200.0, 'palette': ['060606','337663','337663','ffffff']},
'Population Density',0);

# Exposed Population
Map.addLayer(population_exposed, { min: 0, max: 200.0, 'palette': ['yellow', 'orange', 'red']}, 'Exposed
Population');

# MODIS Land Cover
Map.addLayer(LC, { min: 1.0, max: 17.0, 'palette': [
    '05450a', '086a10', '54a708', '78d203', '009900', 'c6b044', 'dcd159',
    'dade48', 'fbff13', 'b6ff05', '27ff87', 'c24f44', 'a5a5a5', 'ff6d4c',
    '69fff8', 'f9ffa4', '1c0dff' ]}, 'Land Cover',0);

# Cropland
croplandVis = { min: 0, max: 14.0, 'palette': ['30b21c']};
Map.addLayer(cropland, croplandVis, 'Cropland',0)

# Affected cropland
Map.addLayer(cropland_affected, croplandVis, 'Affected Cropland');

# Urban
urbanVis = {min: 0, max: 13.0, 'palette': ['grey'], };
Map.addLayer(urban, urbanVis, 'Urban',0)

# Affected urban
Map.addLayer(urban_affected, urbanVis, 'Affected Urban');
Map.setCenter(107.5, 16.6, 11)
Map
```

== Export==

Remove the comments below to export the images to the local drive where

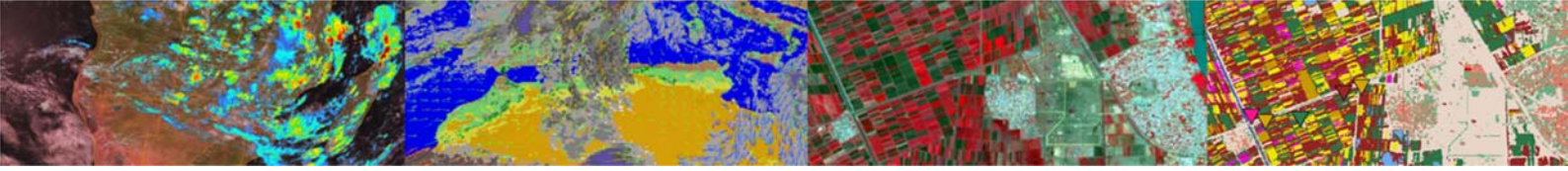


the Jupyter Notebook is located.

Export flooded area as TIFF file
#geemap.ee_export_image(flooded, 'tt-hue_flooded.tif', 200, "EPSG:4326", roi, 1);

Export flood polygons as shape-file
#geemap.ee_export_vector(flooded_vec, 'tt-hue_flooded.shp');

Export auxcillary data as shp?
Exposed population density
#geemap.ee_export_image(population_exposed, 'tt-hue_population_exposed\.tif', 250, "EPSG:4326", roi, 1);



4. Assessment of flash flood impacts in Vinh City

Tran Thuy Chi (ttchi@hunre.edu.vn)

Introduction

In recent years, Nghe An province, including Vinh city, which is located in central Vietnam, is experiencing flooding and is submerged due to number of heavy rains. Changes in precipitation patterns towards more intense storms could lead to widespread flooding. This causes damage to buildings and infrastructure and thus is a significant threat to public safety.

Heavy rains on October 2020 caused losses to property and crops, many houses were inundated. The Sentinel-1 and Sentinel-2 satellites could provide information to map the flood areas.

Objective

The objective of this study is to analyze the effect of 2020 flood to Vinh city and mapping of the inundated areas.

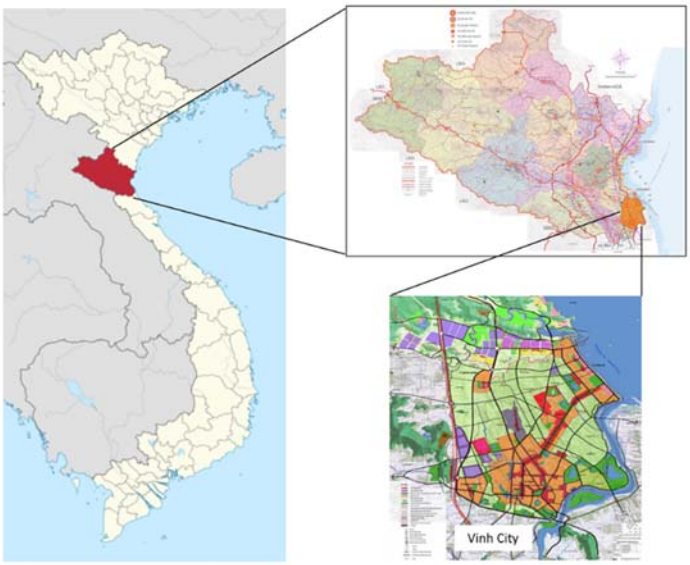


Figure 1. The location of study area

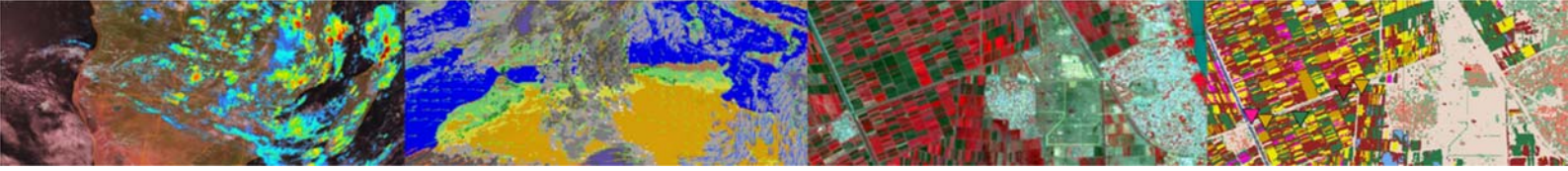
Research question:

1. What was the total submerged area in the flood?
2. Where were submerged areas?

Methodology:

Detection of flood extent immediately after the Molave flood using Sentinel 1.
Software selected: QGIS and Java Script :

- From https://scihub.copernicus.eu/ select images of sentinel 1 for Vinh city
- Apply Google Earth Engine



Results

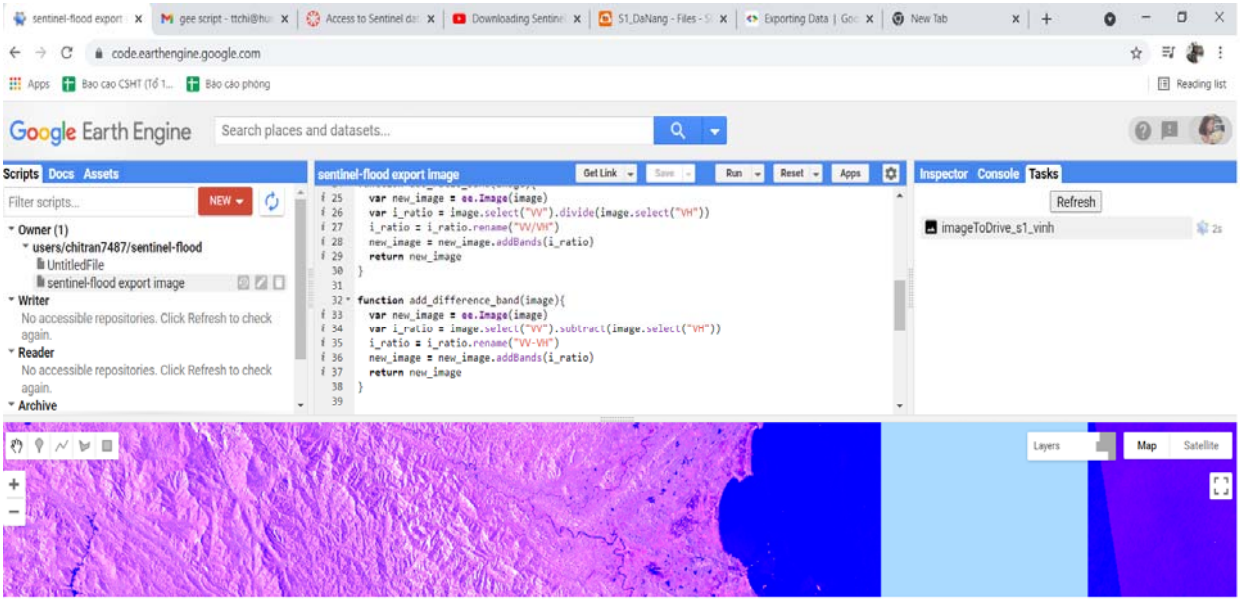


Figure 4. Google Earth Engine Interface

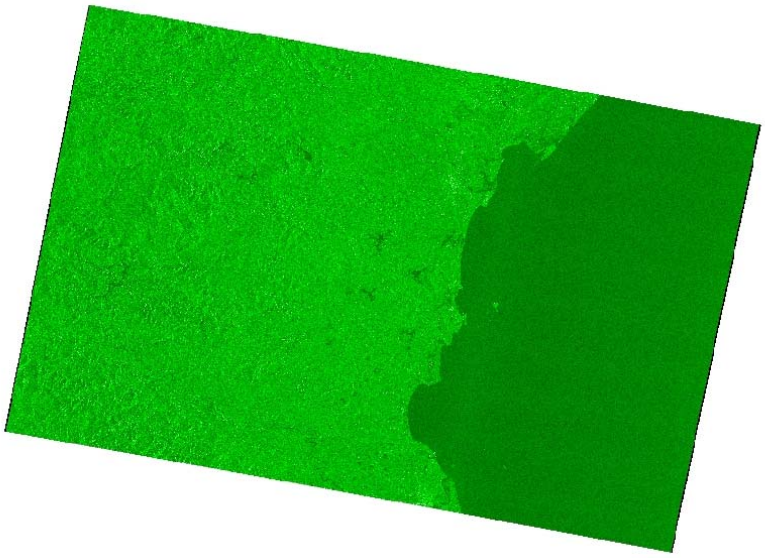


Figure 5. Image of Vinh City, downloaded from Scihub

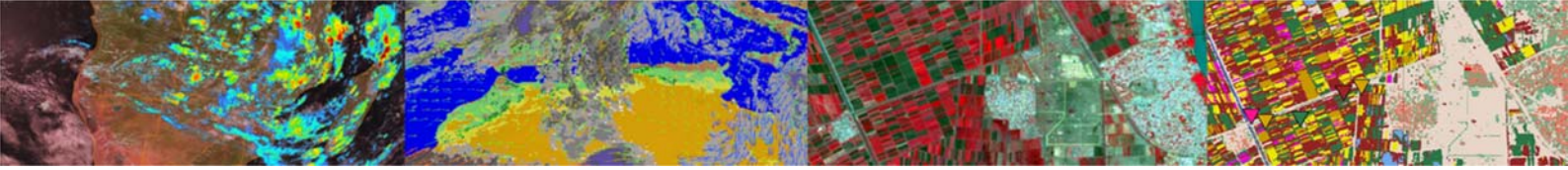
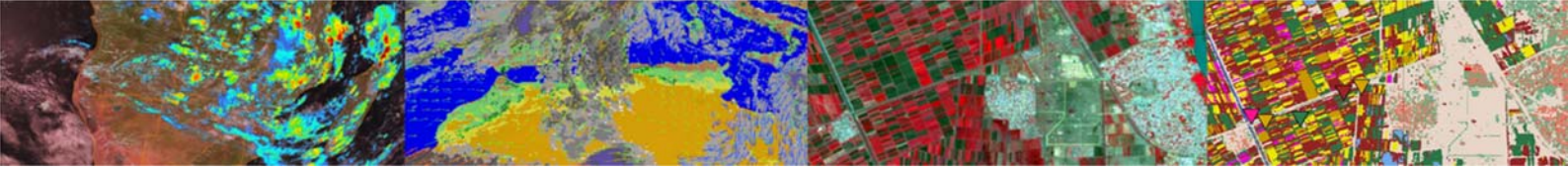
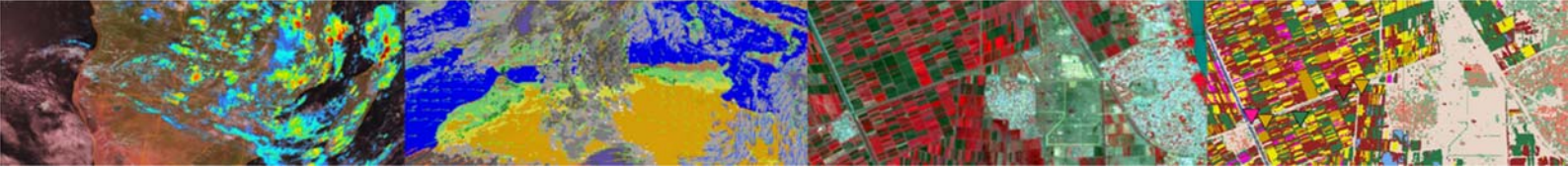


Figure 6. Flooded area in Molave Typhoon



Annex 1: Using Java script to download the image

```
// Java_script_localized_flood_typhoon_Molave_S1

// Define AOI

var geometry = ee.Geometry.Rectangle([105.15, 18.35, 106.0, 19.0]);
var ic = ee.ImageCollection("COPERNICUS/S1_GRD")
var ivp = {"opacity":1,"bands":["VV","VH","VV/VH"],"min":-20,"max":-5,"gamma":1};
var ivp2 = {"opacity":1,"bands":["VV","VH","VV-VH"],"min":-20,"max":0,"gamma":4.935};

// Molave made landfall on October 28 2020
var date_start = ee.Date("2020-10-20")
var date_end = ee.Date("2020-11-10")
var ic_vvvh = ee.ImageCollection('COPERNICUS/S1_GRD')
        .filterDate(date_start,date_end)
        .filterMetadata("transmitterReceiverPolarisation","equals",["VV","VH"])
        .filter(ee.Filter.eq('instrumentMode', 'IW'))
        .map(function(image) {
          var edge = image.lt(-50.0);
          var maskedImage = image.mask().and(edge.not());
          return image.updateMask(maskedImage);
        });
function add_ratio_band(image){
  var new_image = ee.Image(image)
  var i_ratio = image.select("VV").divide(image.select("VH"))
  i_ratio = i_ratio.rename("VV/VH")
  new_image = new_image.addBands(i_ratio)
  return new_image
}
function add_difference_band(image){
  var new_image = ee.Image(image)
  var i_ratio = image.select("VV").subtract(image.select("VH"))
  i_ratio = i_ratio.rename("VV-VH")
  new_image = new_image.addBands(i_ratio)
  return new_image
}
ic_vvvh = ic_vvvh.map(add_ratio_band)
ic_vvvh = ic_vvvh.map(add_difference_band)
var ic_vvvh_desc = ic_vvvh.filter(ee.Filter.eq('orbitProperties_pass', 'DESCENDING'));
var ic_vvvh_asc = ic_vvvh.filter(ee.Filter.eq('orbitProperties_pass', 'ASCENDING'));
Map.addLayer(ic_vvvh_asc,ivp,"S1 ratio [VV,HV,VV/HV]")
Map.addLayer(ic_vvvh_asc,ivp2,"S1 difference [VV,HV,VV-HV]")
Map.addLayer(geometry)
Map.setCenter(105.56,18.68,10)
var select =

// Export the image, specifying scale and region.
Export.image.toDrive({
  image:ic_vvvh_asc.median().toFloat().clip(geometry),
  description: 'imageToDrive_s1_vinh',
  scale:100,
  region: geometry,
  maxPixels: 20000000
});
```



5. Application of remote sensing (Sentinel 2) to build the flood map of the Ca river basin.

Le Thi Thuong, ltthuong.kttv@hunre.edu.vn

Objective

This study will apply observations from Sentinel 2A to construct a the flood map of the Ca river basin during the rainy season.

Research question

How to determine appropriate flooding values for pixels in the Sentinel 2 image?

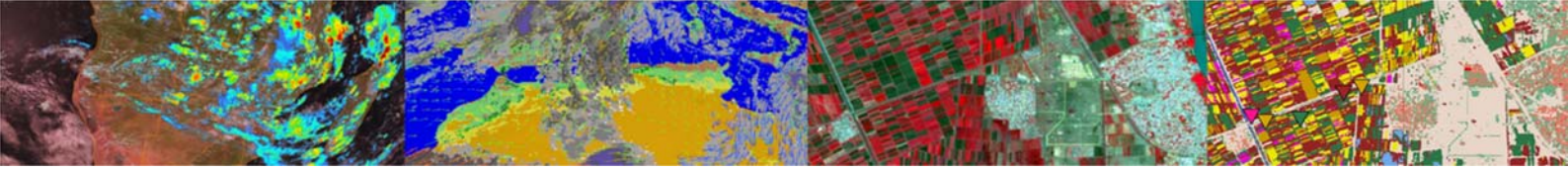
Method and software tool(s) selected

- Earthengine.google.com
- QGIS/SNAP

Data processing and Results obtained

Step 1: The S2A satellite in this study, 10 S2A image scenes were obtained at different times, with Cloud coverage less than 10% is free to download from the ESA website (https://scihub.copernicus.eu/dhus/)

Or Type the word “Sentinel-2” into the search box and open the Sentinel-2 window that appears as shown below:



Sentinel-2: MultiSpectral Instrument (MSI), Level-1C

SENTINEL-2 is a wide-swath, high-resolution, multi-spectral imaging mission supporting Copernicus Land Monitoring studies, including the monitoring of vegetation, soil and water cover, as well as observation of inland waterways and coastal areas.

The SENTINEL-2 data contain 13 UINT16 spectral bands representing TOA reflectance scaled by 10000:

Band	Use	Wavelength	Resolution
B1	Aerosols	443nm	60m
B2	Blue	490nm	10m
B3	Green	560nm	10m
B4	Red	665nm	10m
B5	Red Edge 1	705nm	20m
B6	Red Edge 2	740nm	20m
B7	Red Edge 3	783nm	20m
B8	NIR	842nm	10m
B8a	Red Edge 4	865nm	20m
B9	Water vapor	940nm	60m
B10	Cirrus	1375nm	60m
B11	SWIR 1	1610nm	20m
B12	SWIR 2	2190nm	20m

See Sentinel 2 User Handbook for details. In addition, the following bands are present:

- QA10: currently always empty
- QA20: currently always empty
- QA60: bit mask band with cloud mask information. Bit 10 is set if the corresponding 60m pixel has been marked as

Data availability (time)
Jun 23, 2015 - Nov 2, 2016

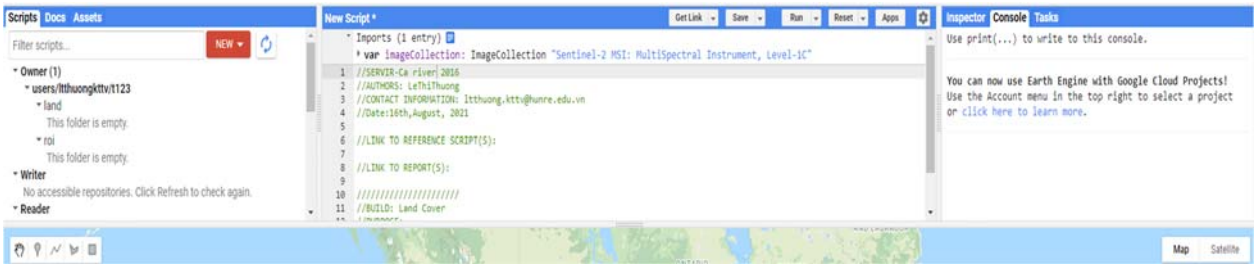
Provider
European Union/ESA/Copernicus

Tags
eu, esa, copernicus, sentinel, msi, radiance

ImageCollection ID
COPERNICUS/S2

Import

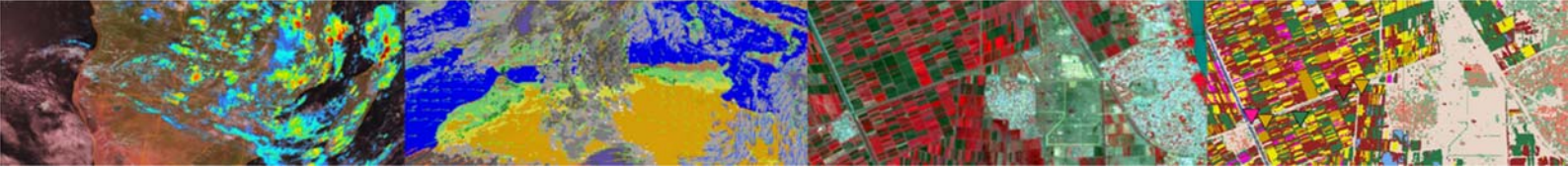
Step 2: I used to Sentinel 2A to import “Earthengine.google.com”.



Step 3: Create template into a new script and fill in the author, date and contact details.

Step 4: Import the fusion table (vector file) of the Ca river basin or your fusion table into the command below:

```
// import the sub basins as a fusion table
var Ca = ee.FeatureCollection('ft:11uQewh4u7NFXneH99YPs6P9F1Au6wxBq959BZWt_',
```



Step 5. Define the interval with the following statement:

```
// filter data and location
var images = s2.filterDate('2016-02-01', '2016-03-31')
                        .filterBounds(Ca);
```

Step 6: Apply below function to remove clouds

```
// cloudMask
function cloudMask(im) {
  // Opaque and cirrus cloud masks cause bits 10 and 11 in QA60 to be set,
  // so values less than 1024 are cloud-free
  var mask = ee.Image(0).where(im.select('QA60').gte(1024), 1).not();
  return im.updateMask(mask);
}

// remove clouds for all images
images = images.???(cloudMask);
```

Step 7: Display images with real color combinations.

// Add to map

Map.centerObject(NgheAn, 8);

Map.addLayer(images.min().clip(NgheAn), {bands: ['B4', 'B8', 'B11'], max: 2048}, 's2 image Feb-Mar 2016');

Step 8: Calculate the Normalized Difference Water Index (NDWI) from the Sentinel image

/Estimate NDVI

var NDVI = NIR.subtract(RED).divide(NIR.add(RED))

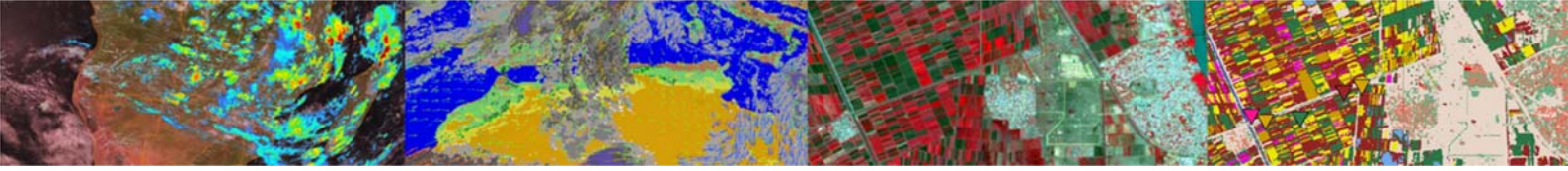
var NDVIc = NDVI.clip(roi)

Map.centerObject(roi,8)

Map.addLayer(NDVIc)

Step 9: Choose the highest value in the set

```
// select maximum NDWI
var s2ndwi = images.select(&amp;quot;NDWI&amp;quot;)
```

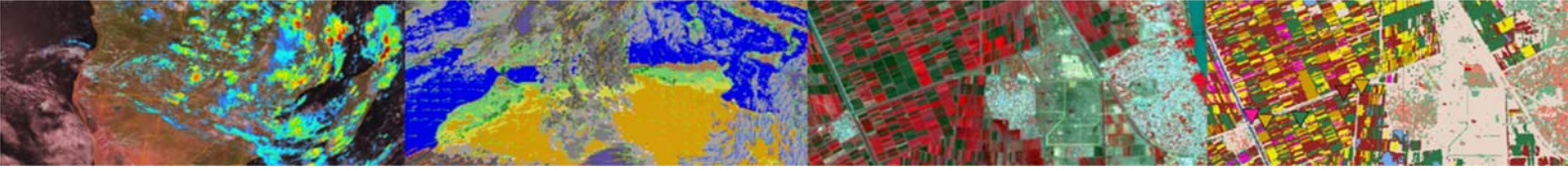


Step 10: Select all pixels with value greater than 0.1

```
// set the threshold
var THRESHOLD = 0.1;
// select pixels greater than threshold
s2ndwi = s2ndwi.gt(THRESHOLD);
```

Step 11: Display the image according to the specified color palette

Concluding remark
I just finished this step. I have done it many times but still have the error that the image cannot be displayed according to the palette (Step 11). I hope you don't mind and guide me about this issue.



6. Building the inundation map for Tich River (Chuong My District, Hanoi) in 2018 to exactly determine the flooded area.

Nguyễn Phương Tú (nptu@hunre.edu.vn/ tuhieukhoi@gmail.com)

Introduction

The studied area is Tich River basin which crosses the districts of Ba Vi, Son Tay, Phuc Tho, Thach That, Quoc Oai, Chuong My. The length of main river is 91km, total basin area of 1330km2.

In 2018, there was a historical flood in this area and at that time, the water level in Tich River was above the warning water level III of 0.22 m, causing the inundation for hamlets of Phu Cao, Thong Dat and Ben Voi, 700 houses, 1800 hectares of rice yard, 120 hectares of secondary crops, etc.

Objective of case study

Images of studied area on 10/7/2018 (before the flood) and 03/8/2018 (after the flood) are taken from Sentinel 1. Then they are analyzed and transferred to Google Earth Engine to visually observe the scope and level of inundation in the studied area.

Preferred processing method

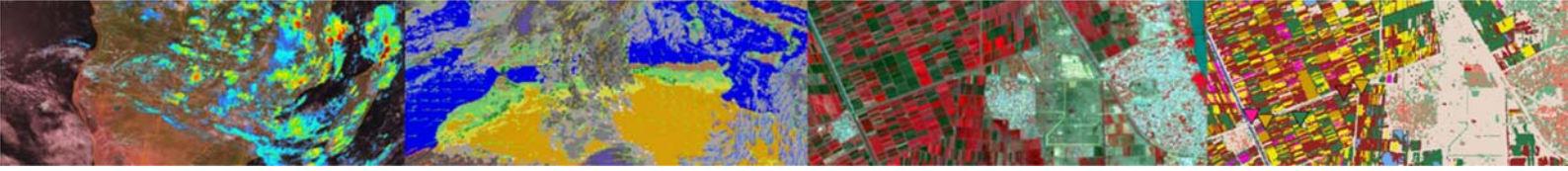
Sentinel 1A in conjunction with Google Earth Format

Methodology

Sentinel 1A is the first satellite in a series of satellites in the Copernicus program, was put into orbit on April 3, 2014. Sentinel-1A is tasked with monitoring ice, oil spills, wind and waves, land use change, terrain deformation, and responding to flood and earthquake emergencies.

Results

- Using the images (before and after the flood) from Sentinel to exactly define the inundation area.
- Pre-processing of the Sentinel 1 data in SNAP



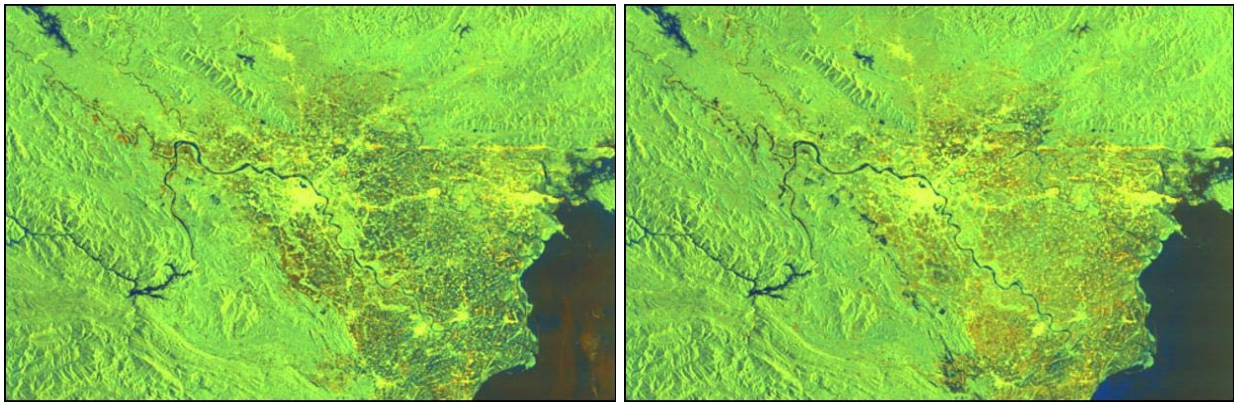
- Comparing two images (on 10/7/2018 and 03/8/2018) to clearly see the flood area
- Comparing the flood area with other layers by converting it to Google Earth format.

Data processing

Using the Sentinel 1A to get the images of Tich area basin on two dates.

Table 1: Images collected for the studying

No.	Studied area	File name	Sensing date	Note
1	Tich area basin	S1A_IW_GRDH_1SDV_20180710T225045_20180710T225110_022738_0276E0_0B91.SAFE	10/7/2018	Before the flood
2		S1A_IW_GRDH_1SDV_20180803T225046_20180803T225111_023088_0281CE_40CF.SAFE	03/8/2018	After the flood



Using SNAP to process these images.

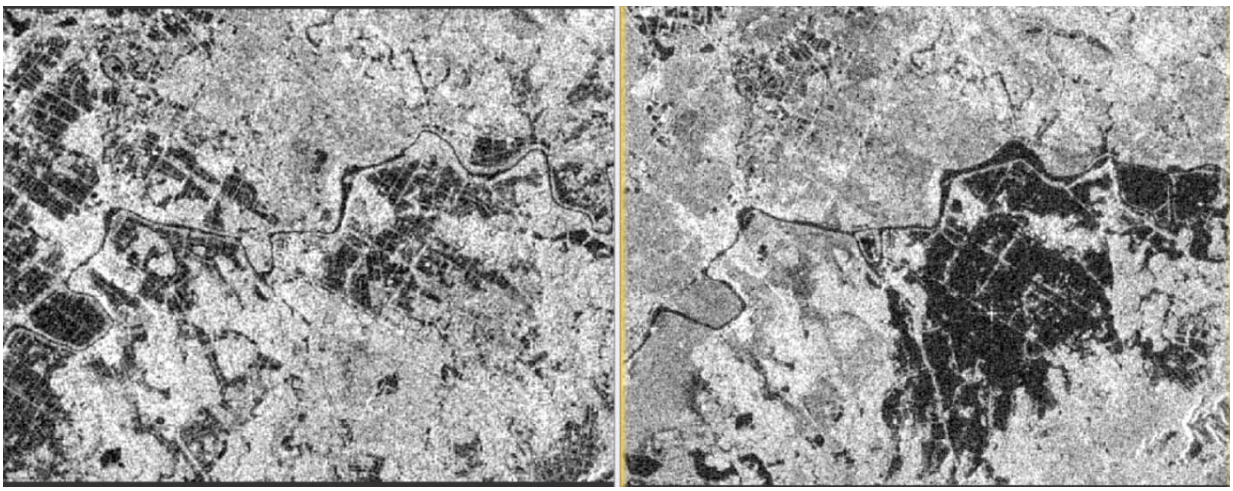
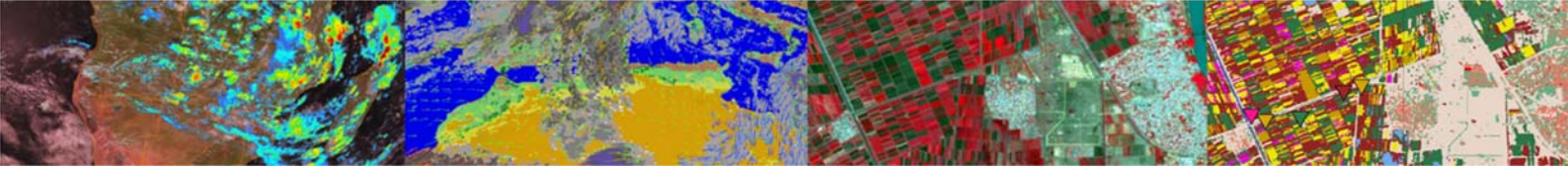
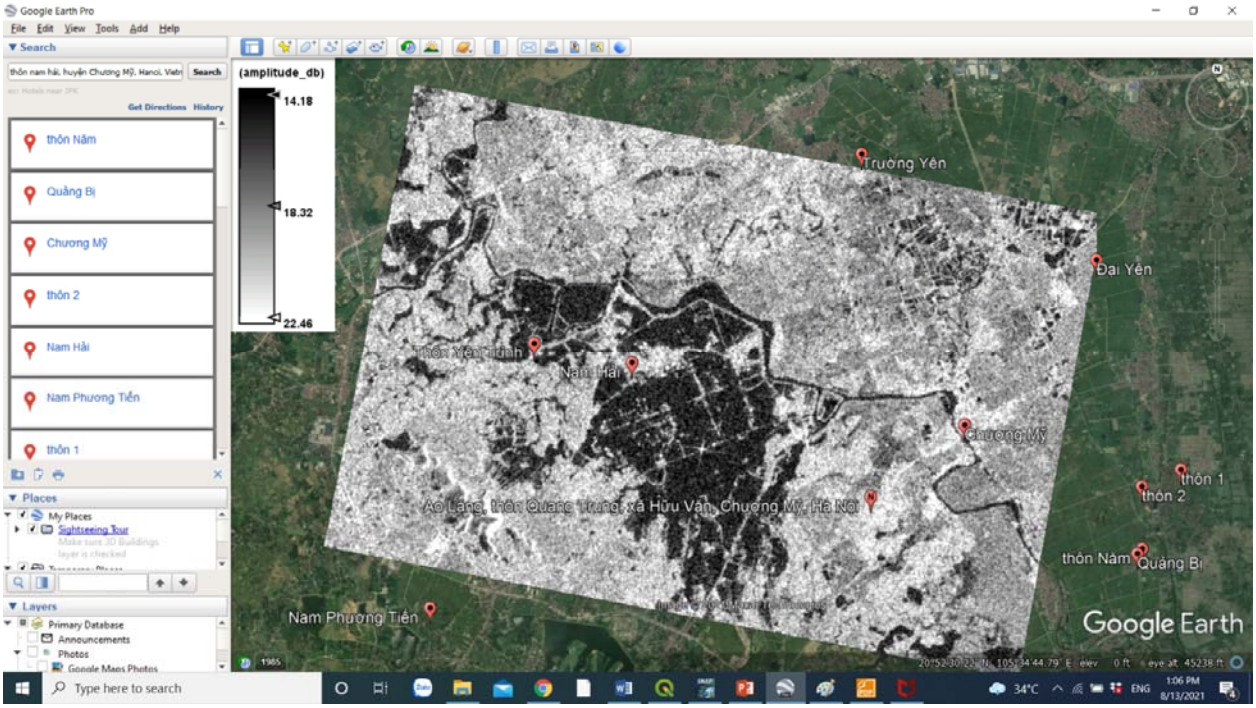


Image of Tich River basin on 10/7/2018 (before the flood)

Image of Tich River basin on 3/8/2018 (after the flood)



Using terrain correction and converting the images into Google Earth KMZ to comparing the flood area with other layers.

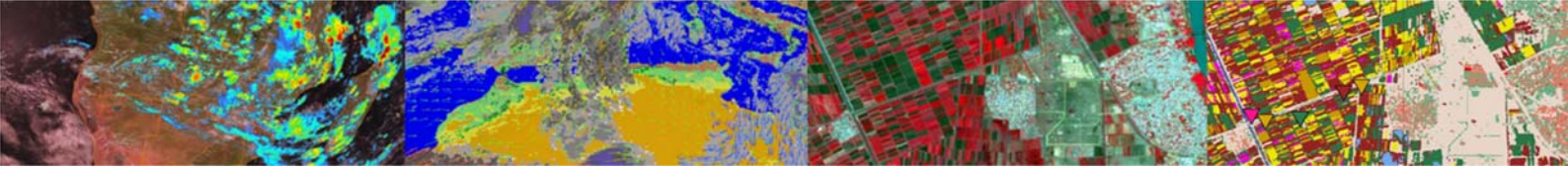


Results obtained:

According to statistic data: In the historic flood, the villages of Nam Hai, Hanh Bo, Nhan Ly and Nam Phuong Tien A commune are the areas which were severely inundated.

The image from Satellite is totally in accordance with the reality. The **black area** includes villages and commune mentioned above.

However, because the visualization of the images is not so good and there is a mixture between the rice paddies and residential areas, it is not too clear to observe the inundation area. For separation of the fields of residential areas, rice paddies, water area and measurement of inundated area, images are applied on QGIS.



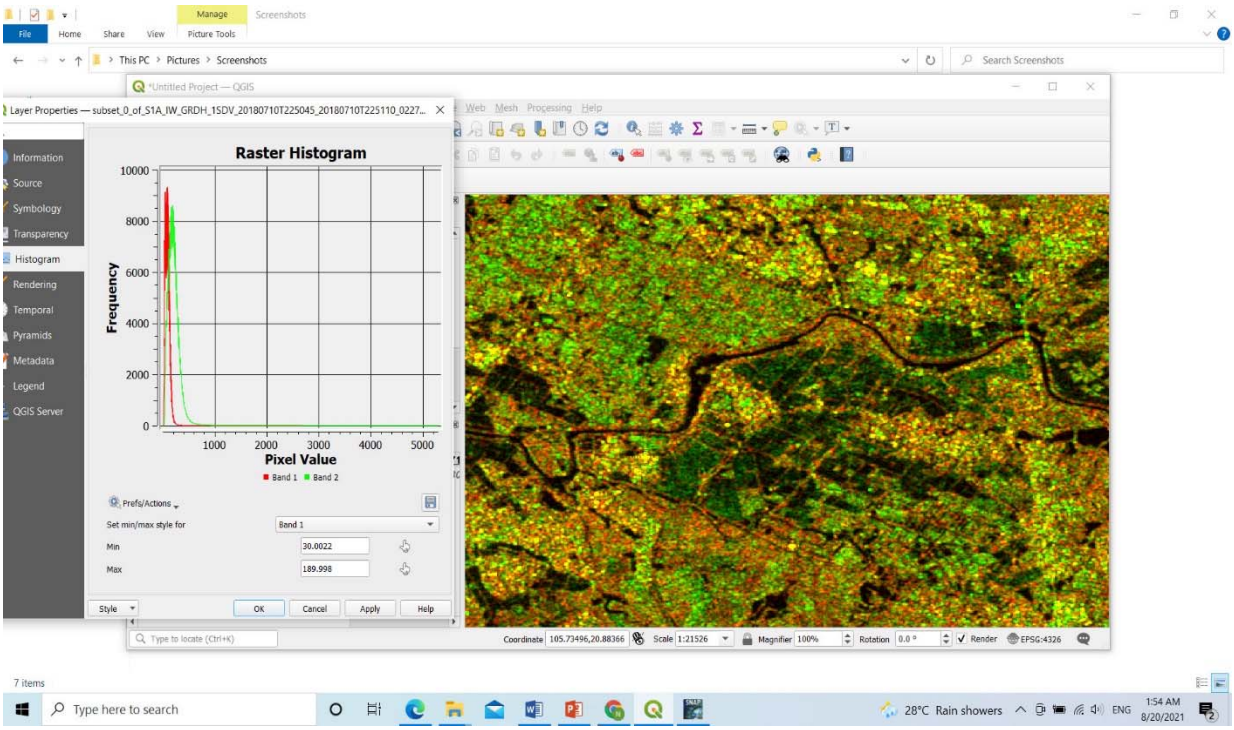


Image of Tich River basin on 10/7/2018

(before the flood)

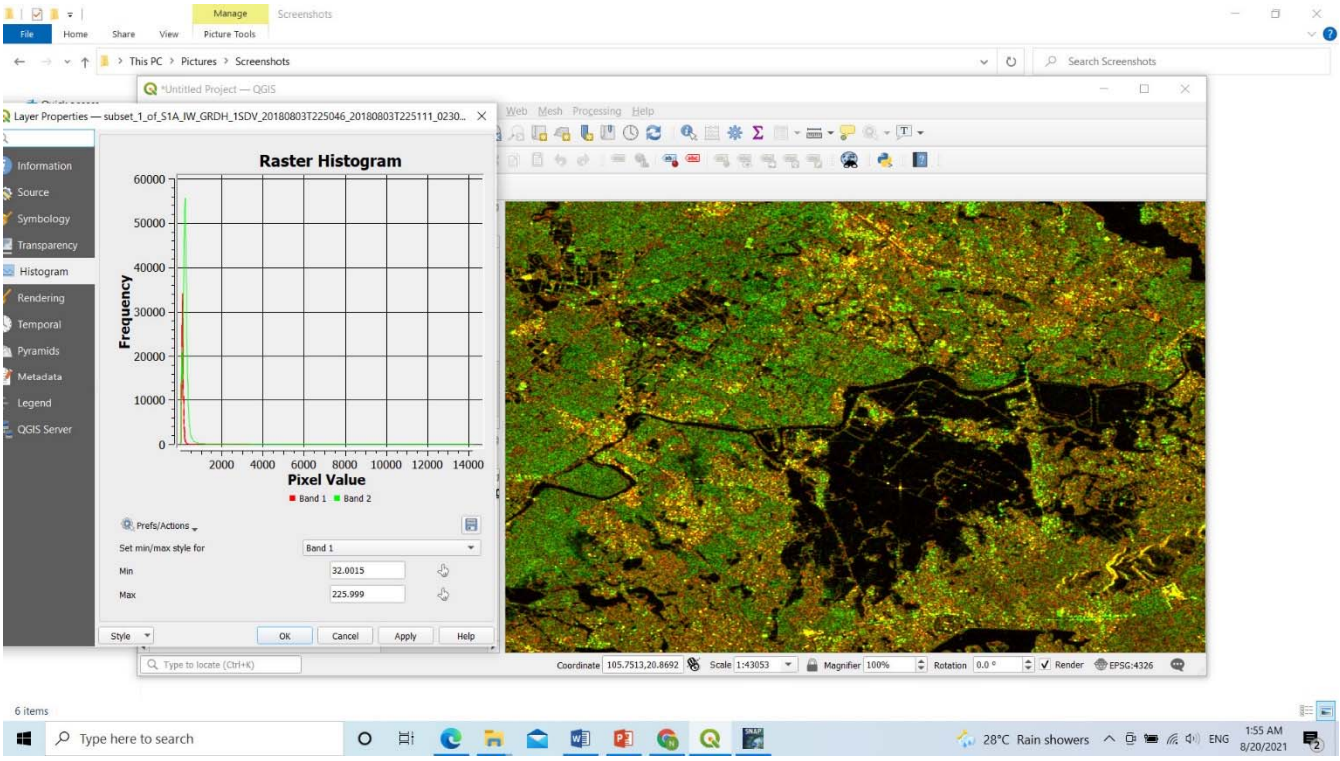
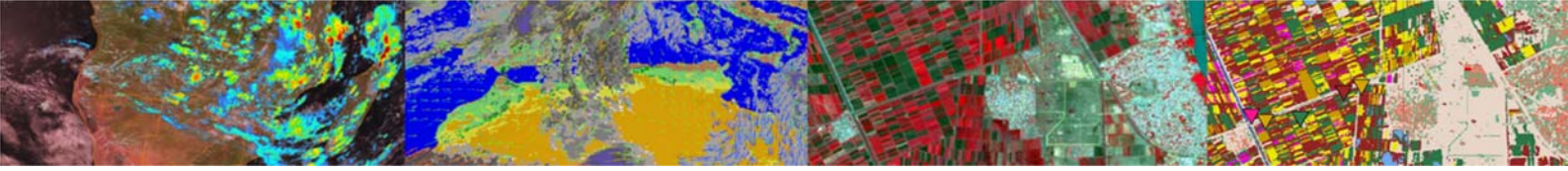


Image of Tich River basin on 10/7/2018

(after the flood)



7. Flood mapping with SAR Sentinel-1 satellite imagery, case study in Ca river basin

Nguyen Tien Quang (ntquang@hunre.edu.vn)

Objective

To determine the flood area based on the Sentinel-1 satellite imagery

Research questions

- How to determine the study time? Typical floods in the basin?
- How to data processing Sentinel 1 image? Tools and methods?
- How to verify the research results compared with the actual flooded area?

Methods

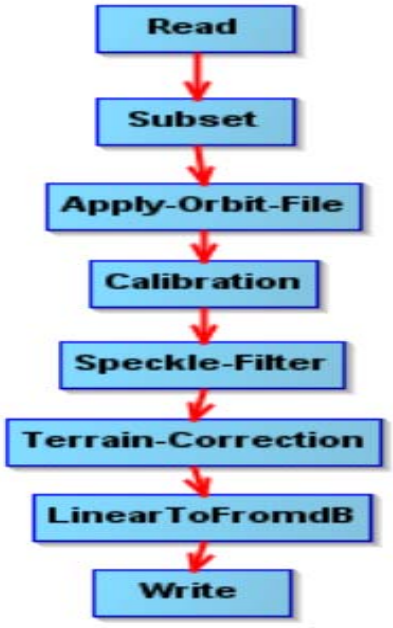
The Sentinel-1, the product by the ESA used to detect the flood shape. The contribution of Sentinel-1 to the application of flood mapping arises from the sensitivity of the backscatter signal to open water (Henry et al., 2006). The sentinel-1 satellite image has four ways of polarized include VV, VH, and HH, and HV. SM, IW, and EW products are available in single (HH or VV) or dual polarization (HH+HV or VV+VH). WV is single polarization only (HH or VV). However, when the calculation to detect flood, band VV or VH will used.

The step by step to detect the water on the Sentinel-1 satellite image include:

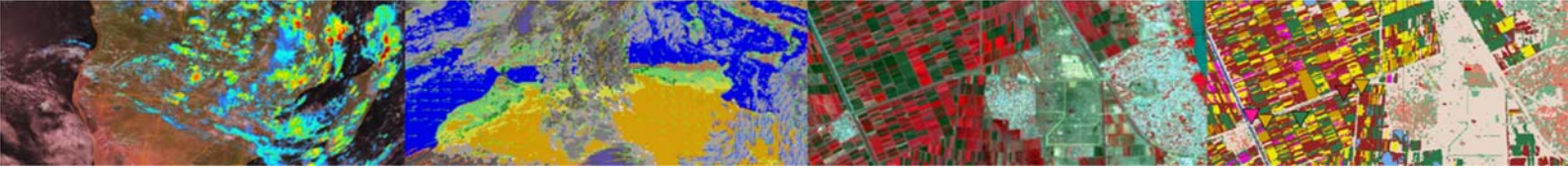
Step 1: Download Sentinel-1 Imagery

Step 2: Download and Install SNAP software

Step 3: Processing the image



• **Subset:** to cut the study area, reduce pixels computation
• **Apply-Orbit-file:** to correct orbit file
• **Calibration:** to convert raw image file to radar backscatter coefficient. Therefore, image calibration has been applied so that the pixel values of the images could directly represent radar backscatter and σ0 bands were created.
• **Speckle-Filter:** to reduce speckle noise and to smooth backscatter data. Because Speckle noise is a form of noise which reduces the quality of an image and may make visual or digital interpretation more difficult.
• **Terrain Correction:** to re-project image from the geometry of the satellite sensors to the geographic projection.
• **LinearToFromdB:** to convert backscatter values from intensity to dB



At Terrain Correction step**,** the algorithm of the range Doppler Terrain Correction was used. In addition, the SRTM with 1 arc-second spatial resolution and bilinear interpolation resampling technique were used for the geometric correction.

Software

To process the Sentinel-1 datasets the following tools are used:

- SNAP
- QGIS

Data processing

The Study Area. In 2020, the central of Vietnam suffered much damage from multi-tropical cyclones. These storms occurred from 2020/10/06 to November combine with heavy rainfall to increase the flood area. Especially, this region has been high slope and a nearby Sea. Therefore, it usually meets the storm every year with high frequency, especially downstream of Ca river basin.

Figure 1: Study location

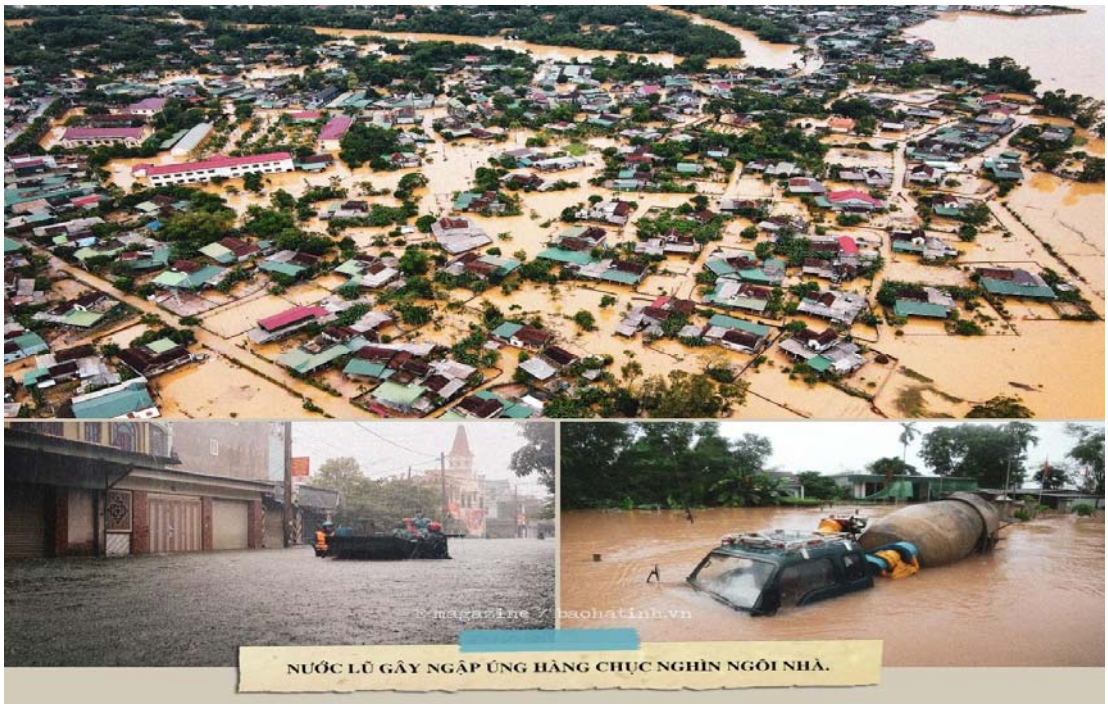


Figure 2: Flood in the Central region cause great damage to people and properties

Data processing

Sentinel -1 images are taken from https://scihub.copernicus.eu/dhus/#/home

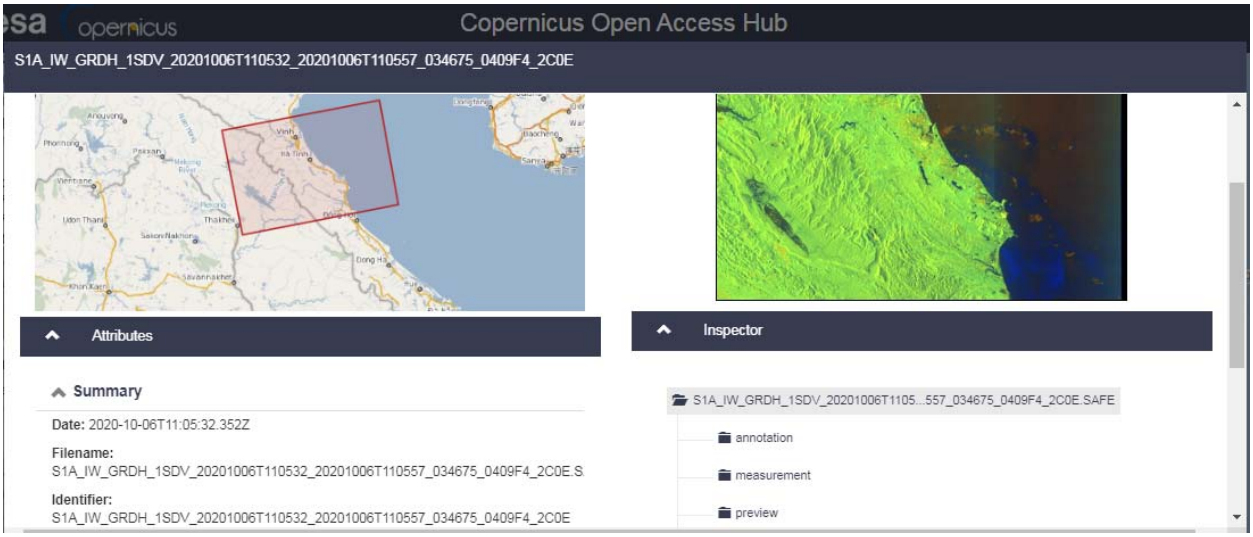
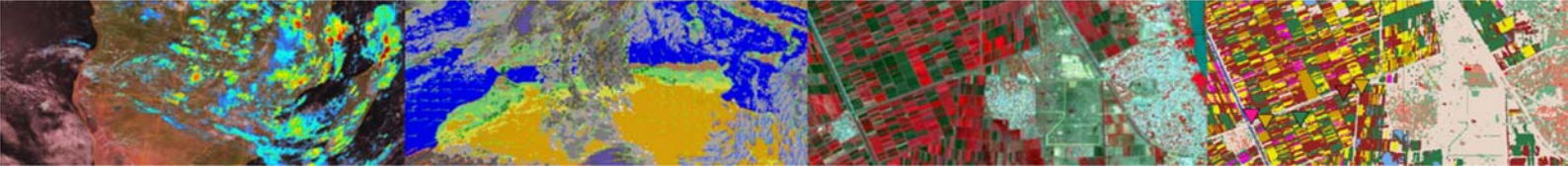


Figure 3: Sentinel -1 image of study area

Table 1: Database of research

No.	Time	Imagery file name	Imagery source
1	06/10/2020	S1A_IW_GRDH_1SDV_20201006T110532_20201006T110557_034675_0409F4_2C0E	https://scihub.copernicus.eu/dhus/#/home
2	18/10/2020	S1A_IW_GRDH_1SDV_20201018T110532_20201018T110557_034850_041018_E503	https://scihub.copernicus.eu/dhus/#/home

Processing and screenshots are based on ESA's Sentinel Application Platform (SNAP) version 8.0 64-bit, with the method described above.



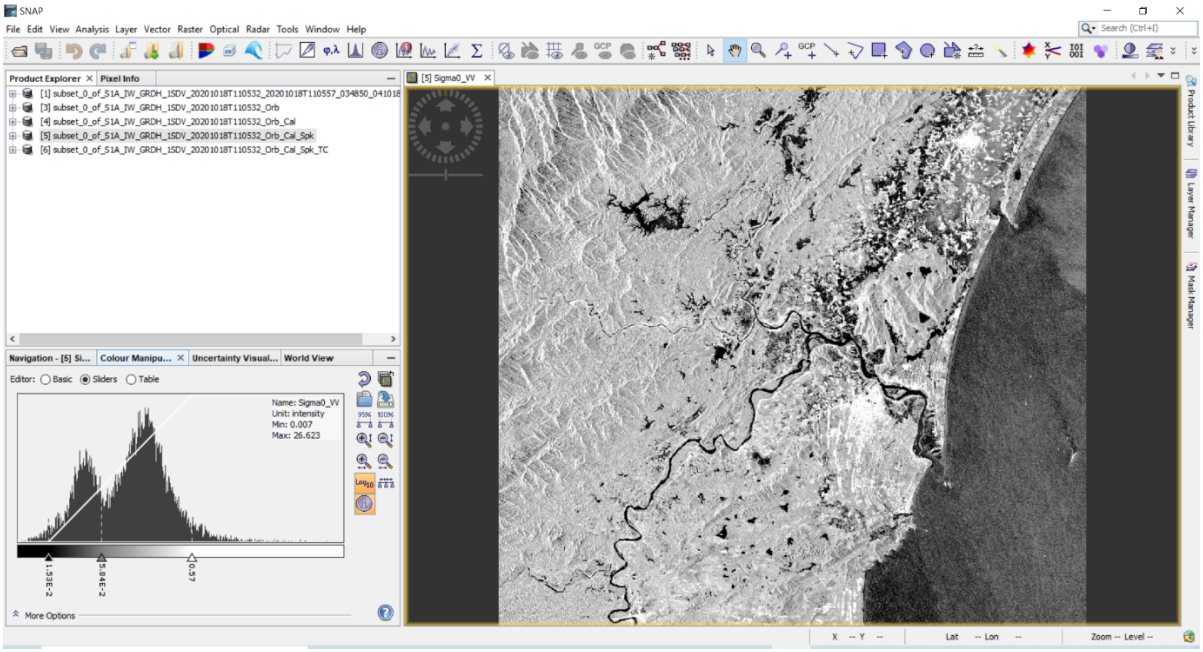


Figure 4: Data processing with SNAP

The post flood image will be used to map the flood using QGIS. Based on the map, user can see and observe the change of the inundation area between begin and after flood at Ca river basin downstream.

Results

Some challenges have been encountered in defining flood zones using SAR in this study. These situations are radar ball, smooth surface, and steep incidence angle. Radar shadows can confuse flooded areas because shadow areas do not provide backscatter as they would appear on smooth water.

The Central Flood of 2020 or storm surge, caused flooding throughout Central Vietnam, starting from the night of October 6, dawn on October 7, 2020 to early December 2020. The first flood from October 6 to 13, the central provinces were seriously affected from damage to facilities and infrastructure to damage to people and their lives. The second flood from October 16, the central region continues to be affected by the new tropical depression in the process of turning into a storm with cold air, constantly receiving heavy rains, prolonged floods (https://vi.wikipedia.org). The data of the study consists of 2 waves, October 6, 2020 and October 18, 2020, corresponding to the time of flood onset and the time of strong flood activity in the Central region.

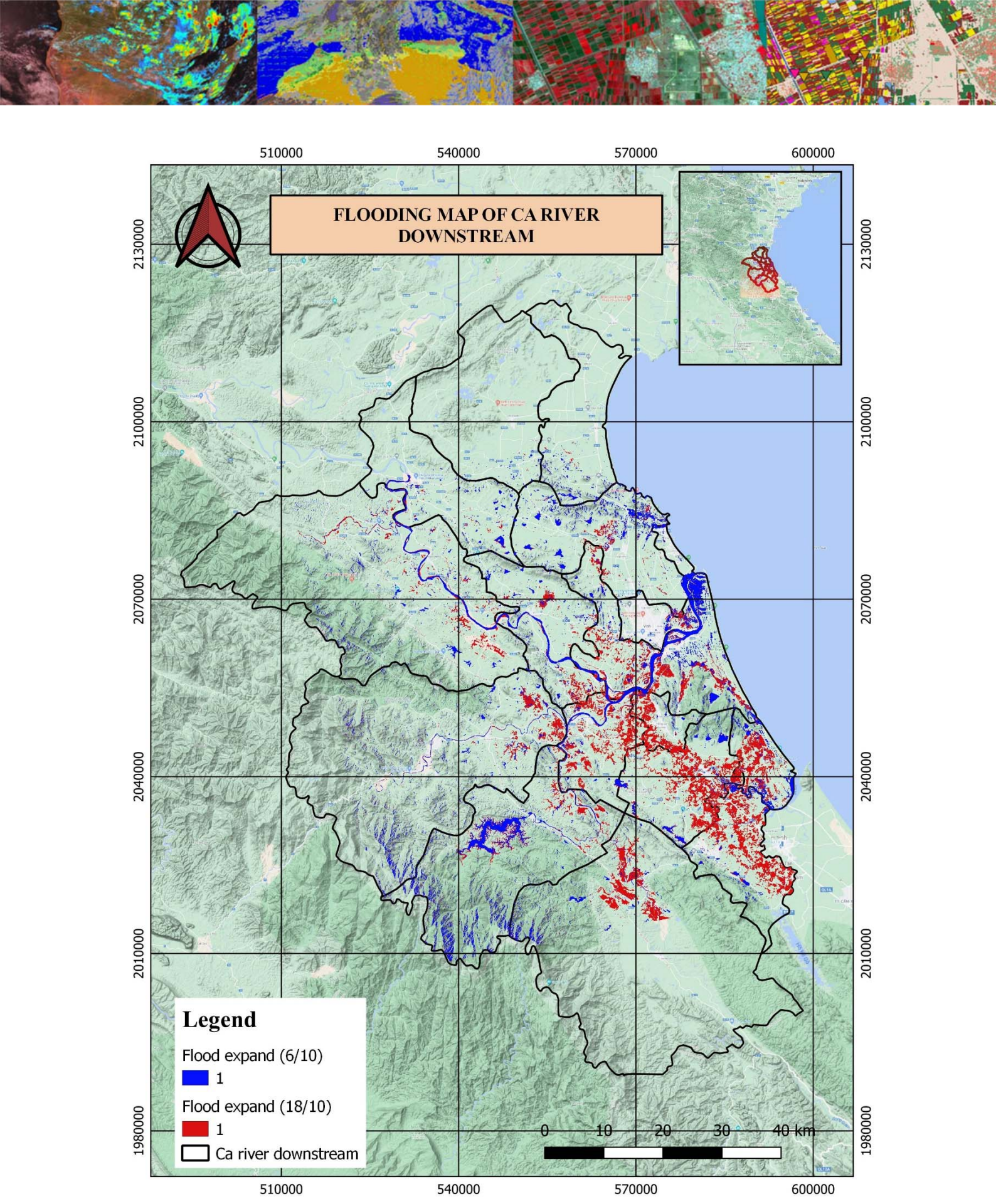
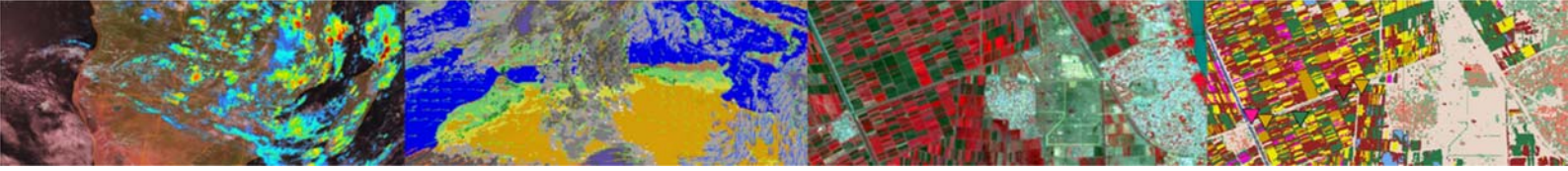


Figure 5: Flood mapping of Ca river basin downstream (6/10 & 18/10)

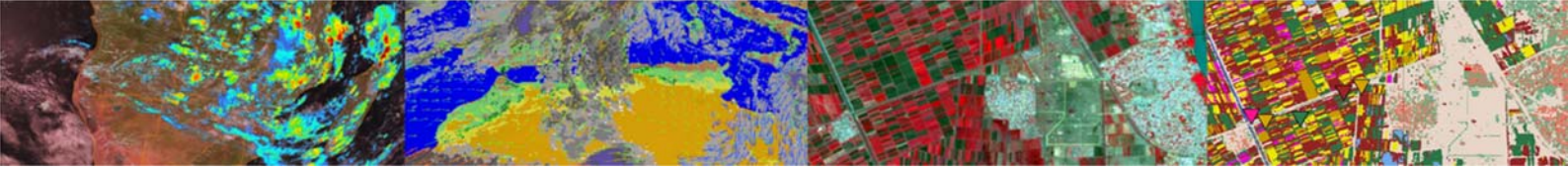


Data of October 6, 2020:

The results show that these regions did not occur more water . This amount of water to include at the river, pond, and holes, etc. The calculated water surface area is 292,35 km^2.

Data of October 18, 2020:

The value of the inundation area on 18 October 2020 is very high. Owing to, from 16 to 19 October, some areas in Nghe An, Ha Tinh province received heavy rainfall (from Tropical depression Ofel). The calculated water surface area is 581,4 km^2.



8. Application of Sentinel satellite for the Thua Thien Hue province flooding October in 2020

Nguyen Thi Bich Ngoc (ntbngoc@hunre.edu.vn)

Introduction

Flooding in Central Vietnam in 2020 is a historic flood that occurred in Central Vietnam. Flooding starts from October 7, 2020 to the beginning of December 2020. Flooding, landslides occur in many places.

During the period of October and November 2020, the circulation of tropical depressions and monsoons caused many localities in the Central region to be flooded on a large scale, with floodwaters rising, dividing areas.

Flood to alarming level IV, belonged to the class of dangerous natural disasters, biggest risks of Vietnam. it had far-reaching effects and impacts causing damage and damage to the whole region, destroying, delaying and pushing socio-economic conditions of Central Vietnam, especially in provinces such as Quang Binh, Quang Tri, Thua Thien Hue, Da Nang, Quang Nam and Quang Ngai.

Thua Thien Hue is one of the provinces most affected by this flood. The whole province has 53,385 houses flooded from 0.2 - 1.2m, some places are higher. On the afternoon of October 11, it was reported that this locality had 3 persons died, 1 missing and 7 injured due to flooding.

Flooding Hue City

Horizontal water flooded houses

Objectives

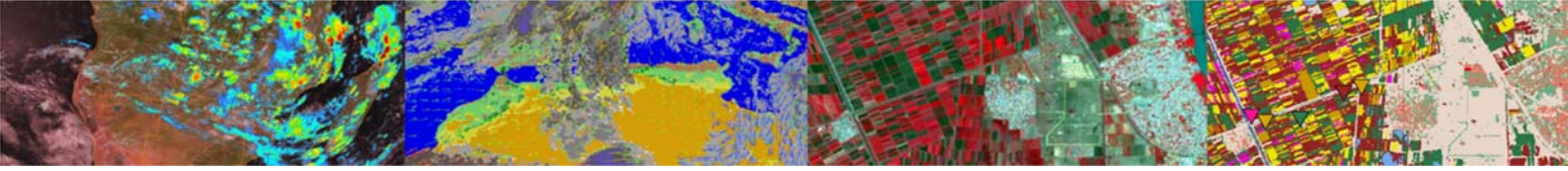
- Start familiar with the satellite datasets, data sources the Sentinel products.
- To be familiar with the software and tools for Sentinel imagery processing for research in Thua Thien Hue province flooding October in 2020

Research questions

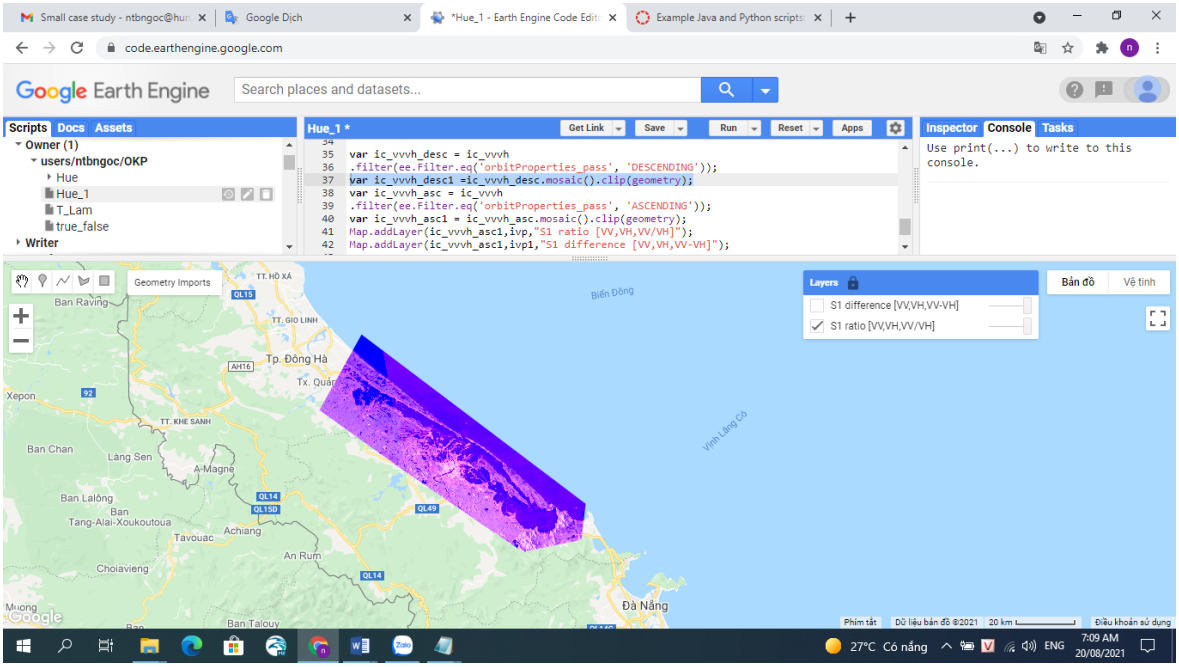
- What is the largest flooded area?
- Which area is most affected?

Method and software

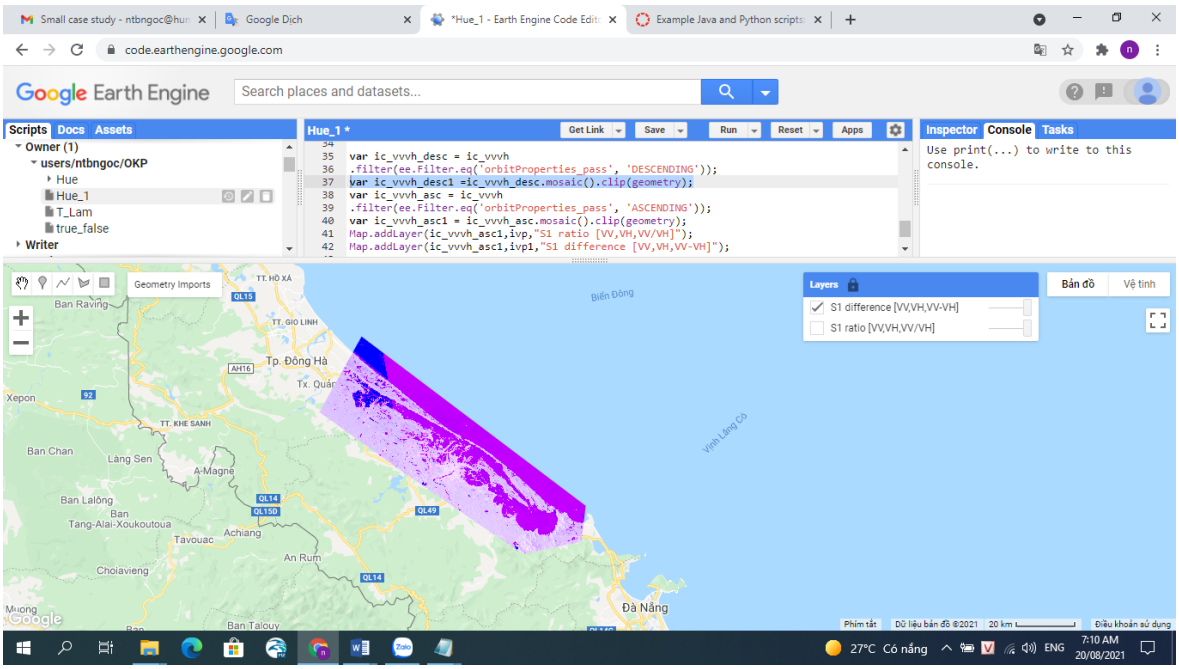
Processing Sentinel 1 images using Google Earth Engine and JavaScript



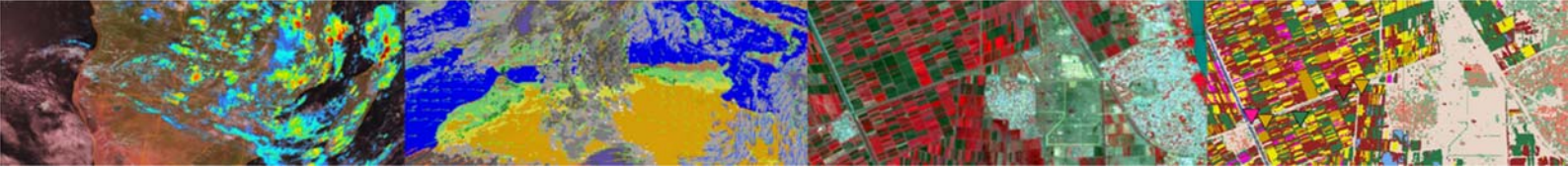
Results



Layer Sentinel 1 ratio

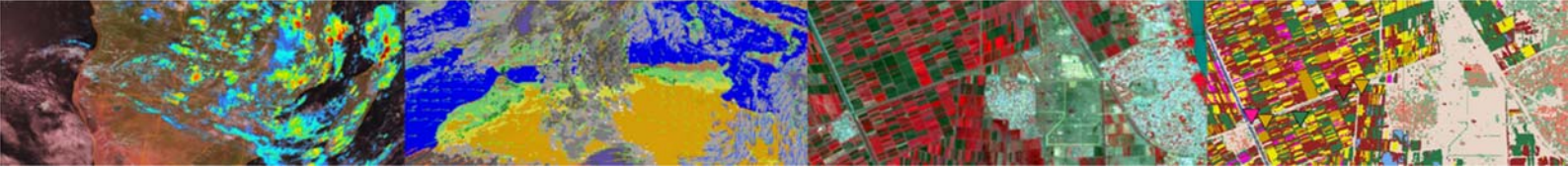


Layer Sentinel 1 difference

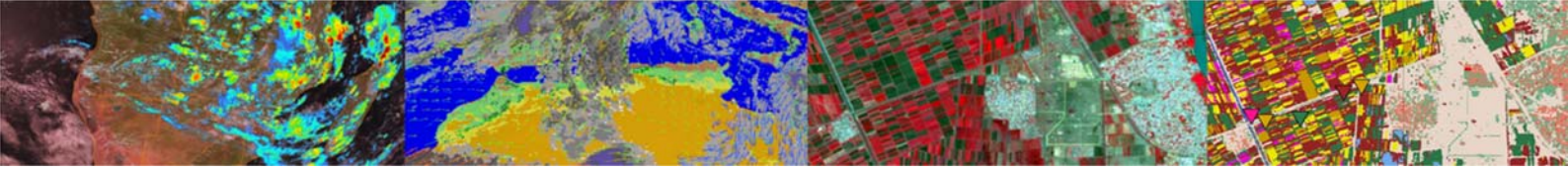


Annex 1: JavaScript

```
//select my own study area
var geometry: Polygon, 7 vertices
type: Polygon
coordinates: List (1 element)
0: List (8 elements)
0: [107.3599259738554,16.813108784372094]
1: [107.32422040744915,16.665817978648487]
2: [107.4807755832304,16.52368011451461]
3: [107.67028974338665,16.31554930622435]
4: [107.81311200901165,16.217993988220325]
5: [107.99987958713665,16.254912510001017]
6: [108.0136124972929,16.365626411653093]
7: [107.3599259738554,16.813108784372094]

//set time frame
var date_start = ee.Date("2020-10-20");
var date_end = ee.Date("2020-11-10");

//refer script Java_script_localized_flood_typhoon_Molave_S1
var ic = ee.ImageCollection('COPERNICUS/S1_GRD');
var ivp = {"opacity":1,"bands":["VV","VH","VV/VH"],"min":-20,"max":-5,"gamma":1};
var ivp1 = {"opacity":1,"bands":["VV","VH","VV-VH"],"min":-20,"max":3,"gamma":4.935};
var ic_vvvh = ee.ImageCollection('COPERNICUS/S1_GRD')
.filterDate(date_start,date_end)
.filterBounds(geometry)
 .filter(ee.Filter.eq('resolution_meters',10))
.filterMetadata("transmitterReceiverPolarisation","equals",["VV","VH"])
.filter(ee.Filter.eq('instrumentMode','IW'))
.map(function(image) {
  var edge = image.lt(-50.0);
  var maskedImage = image.mask().and(edge.not());
  return image.updateMask(maskedImage);
});
function add_ratio_band(image){
  var new_image = ee.Image(image);
  var i_ratio = image.select("VV").divide(image.select("VH"));
  i_ratio = i_ratio.rename("VV/VH");
  new_image = new_image.addBands(i_ratio);
  return new_image;
}
function add_difference_band(image){
  var new_image = ee.Image(image);
  var i_ratio = image.select("VV").subtract(image.select("VH"));
  i_ratio = i_ratio.rename("VV-VH");
  new_image = new_image.addBands(i_ratio);
  return new_image;
}
ic_vvvh = ic_vvvh.map(add_ratio_band);
ic_vvvh = ic_vvvh.map(add_difference_band);
var ic_vvvh_desc = ic_vvvh
.filter(ee.Filter.eq('orbitProperties_pass', 'DESCENDING'));
var ic_vvvh_desc1 =ic_vvvh_desc.mosaic().clip(geometry);
var ic_vvvh_asc = ic_vvvh
```



```
.filter(ee.Filter.eq('orbitProperties_pass', 'ASCENDING'));
var ic_vvvh_asc1 = ic_vvvh_asc.mosaic().clip(geometry);
Map.addLayer(ic_vvvh_asc1,ivp,"S1 ratio [VV,VH,VV/VH]");
Map.addLayer(ic_vvvh_asc1,ivp1,"S1 difference [VV,VH,VV-VH]");
```



9. Land cover classification from Sentinel 2 data for Can Tho Province, Mekong Delta Vietnam

Pham Van Tuan (phamvantuan@ait.asia)

Introduction

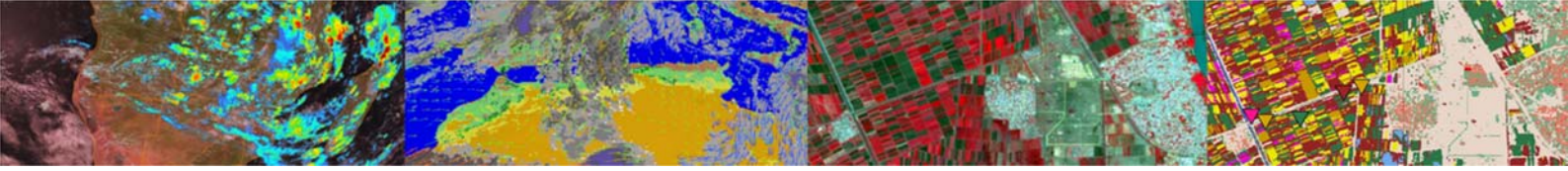
The Mekong Delta is the third largest delta in the world, and a globally recognized agricultural production zone and biodiversity hotspot. In recent decades, the Mekong Delta land cover is rapidly changing due to anthropogenic activities (e.g., agricultural expansion and urbanization) and natural processes (e.g., flooding) (Li et al., 2020). The development of satellite remote sensing technology has revolutionized the approaches in monitoring the natural and human resources on the Earth's surface, and this technology makes it possible to monitor large areas. These satellites produce different remotely sensed data for different applications, such as forest, urban, natural hazard, and agricultural monitoring (Jucker et al., 2017). Since the launch of the first Sentinel satellite in 2014, the Copernicus Programme, in collaboration with the European Space Agency (ESA), provide high-resolution satellite data for land cover/use monitoring, climate change and disaster monitoring (Malenovský et al., 2012). The scientific community, government agencies, and private sectors have used Sentinel-2 data for different applications, such as agricultural, urban development, and forest monitoring. There have been many studies based on Sentinel-2 data since the launch of these satellites in 2015. Many studies have also reported the superiority of Sentinel-2 over similar sensors such as Landsat-8. The application of Sentinel-2 data differs from region to region, especially the type of data which is integrated with Sentinel-2 (Phiri et al., 2020).

The purpose of the research on land use/cover classification in Can Tho city, Mekong Delta Vietnam using Sentinel 2 to investigate, implement the use of Sentinel 2 satellite data using two classified methods including the Classification And Regression Tree (CART) and Random Forest method to produce land cover estimates that achieve 95% overall accuracy.

Material

Study area

Can Tho City lies in a tropical, monsoonal climate with defined seasons: rainy, from May to November, and dry, from December to April. Average annual humidity is 83%, annual rainfall varies between 1,500 - 1,800 mm per year (with more than 90% of rainfall occurring during the rainy season) and a yearly average temperature of 27 °C (Dragon Institute, 2009). Can Tho is located along the south-west side of Hau (Basac) River, which is one of the two main branches of the Mekong River flowing in Vietnamese territory. Each year, the Hau River carries more than 200 million m^3 of freshwater from upstream, i.e. the Mekong River's catchment in China, Myanmar, Laos, Thailand, and Cambodia, to the sea. In the period of September – October, the average water flow in Hau River reaches 15,000 - 16,000 m^3/s, and the flooding season is usually accompanied by bank erosion. In contrast, during dry season, the river flow drops to 1,600 - 1,700 m^3/s leading to water shortages (Dragon Institute, 2009) and increasing risk of salinity intrusion (CNREM and DONRE, 2009)



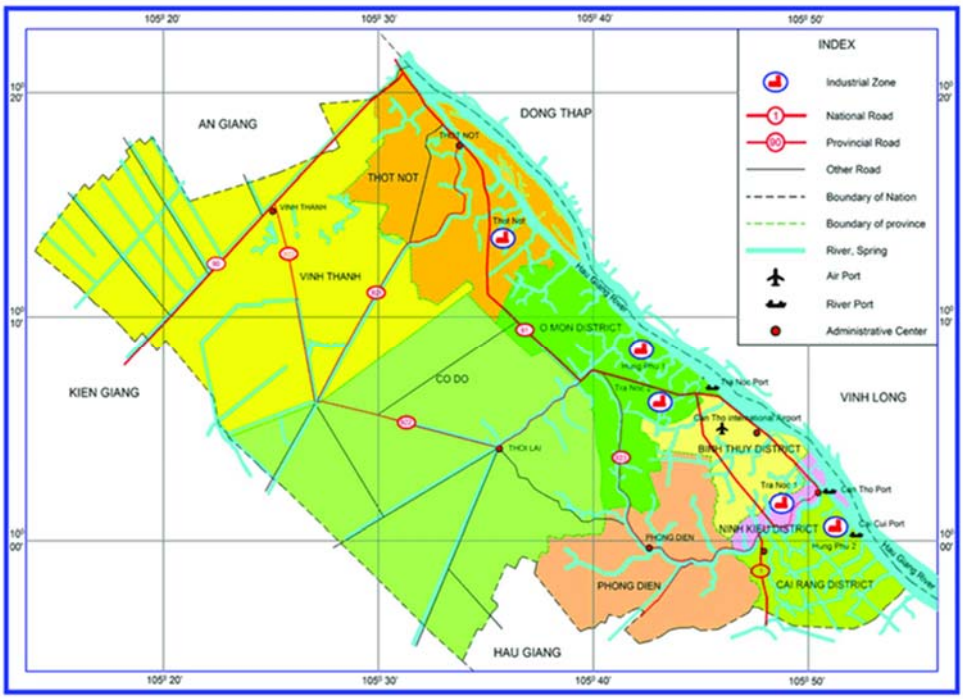


Figure 7 - A map of Can Tho city

Methods

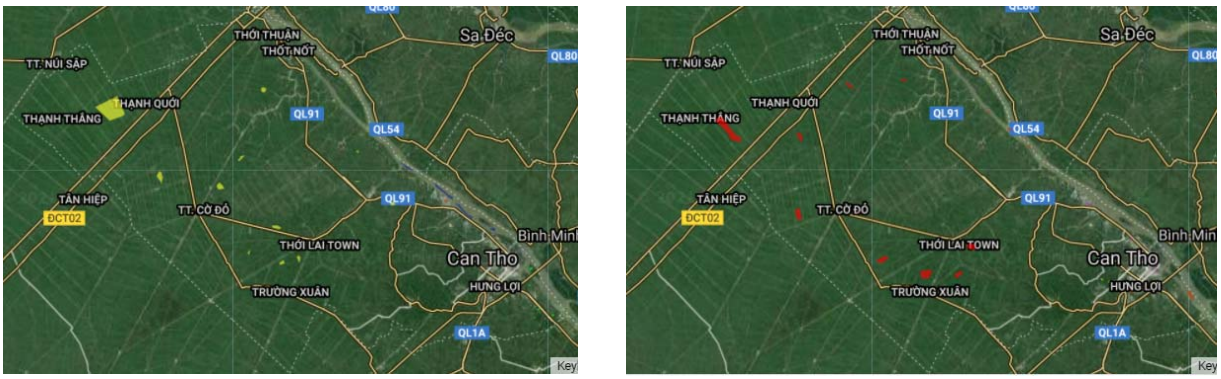
Google Earth Engine will be applied to import and analyze Sentinel 2A image in the dry season 2020. A JAVA scripts will be written to determine various vegetation indices (NDVI, SAVI) and Normalized Difference Water Index (NDWI). Then, the Classification and Regression Trees and Random Forest are applied to train the classifier and classify land use. Raster Analysis Tools in QGIS is used to develop land cover map and estimate area and percentage of each land cover type.

Data processing

The classification of land use in Can Tho city used Sentinel 2 satellite data with 10 m spatial resolution obtained from March 2020 to June 2020 as a representation of the dry month.

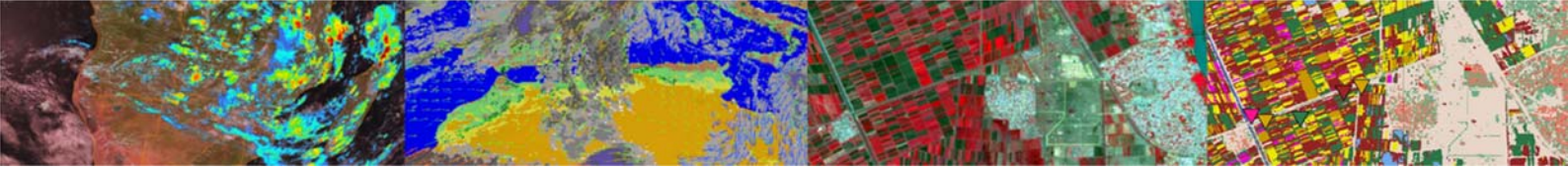
Determination of Land Use Classes, Sample and Sample Training

The training sample data includes 5 classes, as follows: (1) urban, (2) paddy, (3) fruits and others, and (4) water surface (river, lake). Polygon will be created to identify all designated land cover classes. The sample is taken separately for training and data validation



a) Training sample for classification b) Training sample for validation

Figure 8 - Training and test Area



Classification methods

Classification and Regression Trees or CART for short is a term introduced by Leo Breiman to refer to Decision Tree algorithms that can be used for classification or regression predictive modeling problems.

The Random Forest method has several advantages such as more efficient results for large data, can handle thousands of variables without reducing it, gives an estimate of variables that are important in classification, can produce unbiased estimates of generalization errors, computationally lighter than other tree ensemble methods.

Results

Figure 3 and Figure 4 shows the spatial distribution of the 4 land cover types including urban, paddy, fruits and water surface in Can Tho city in the dry season of 2020. The validation results show a higher accuracy in applying Random Forest method to classify land cover in Can Tho city, with accuracy value approximately 0.95. The overall accuracy of CART method is 0.91.

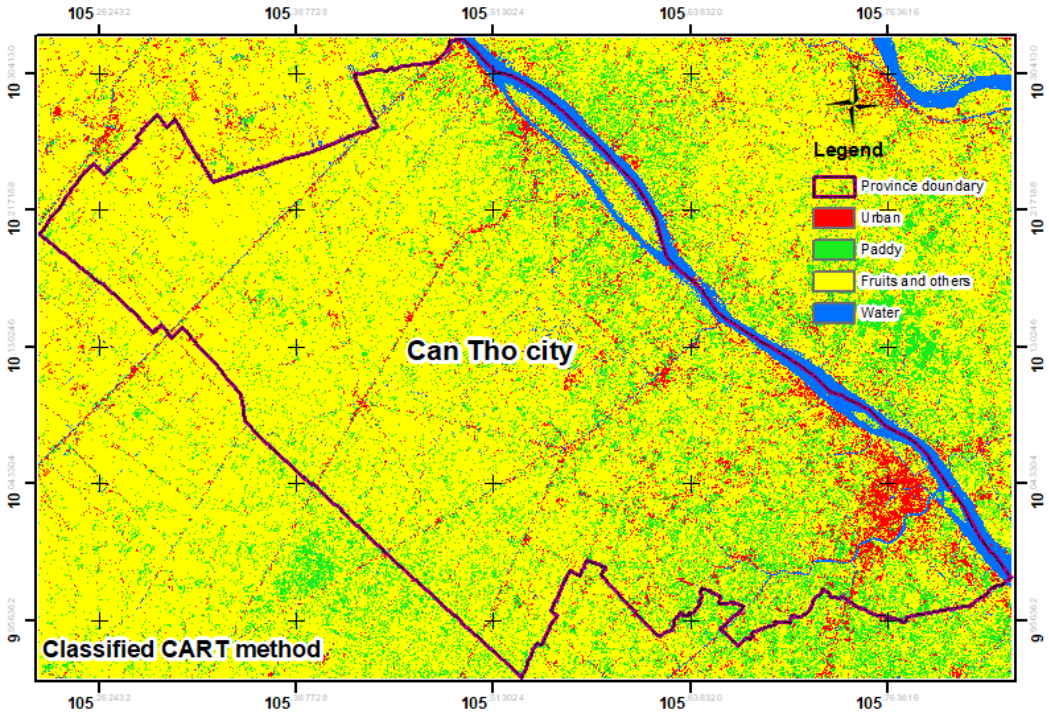
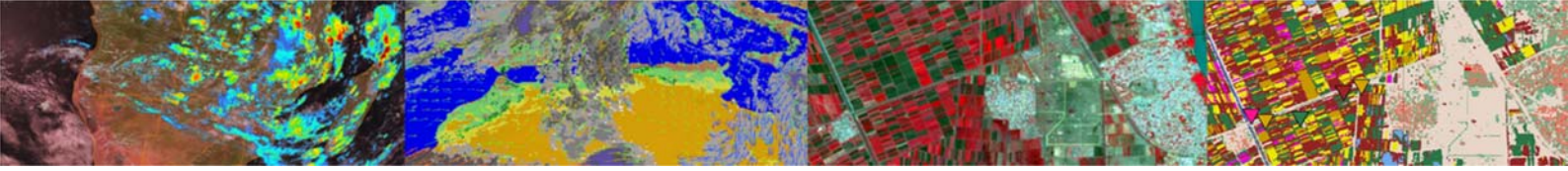


Figure 9 – Map of land cover in Can Thor city by using CART method



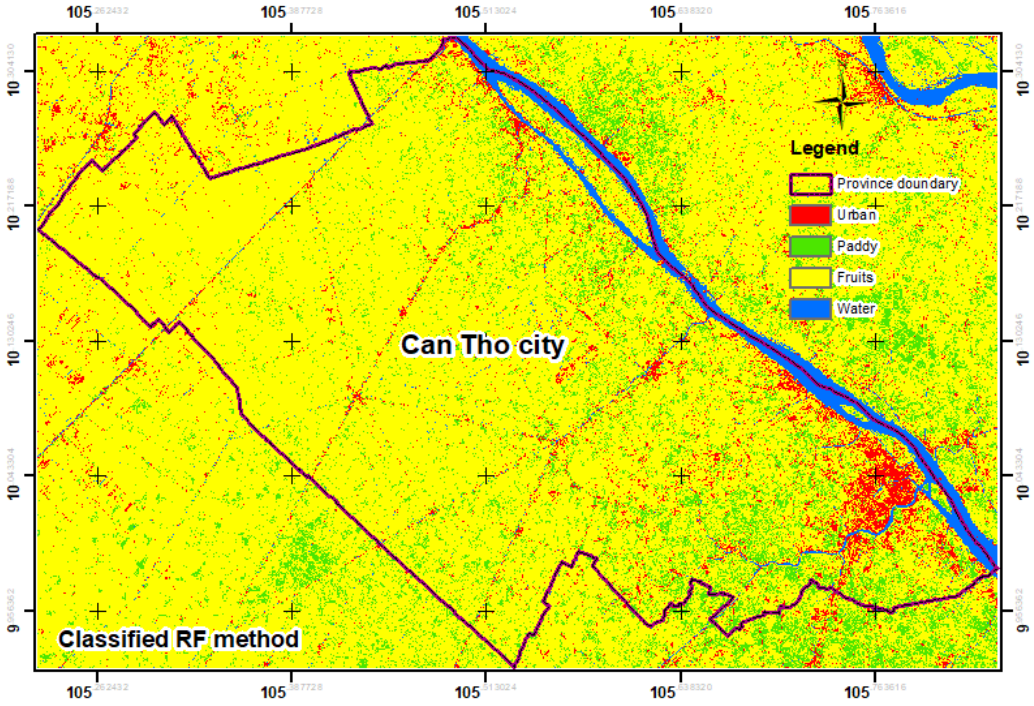


Figure 10 - Map of land cover in Can Thor city by using RF method

Percentage of paddy area by using RF method is about 81%, and higher than the one by using CART method by around 5%. In contrast, there is a lower percentage, about 4% in area of fruits. Urban and water surface area show a similar value in both CART and RF method, over 5% and 3%, respectively.

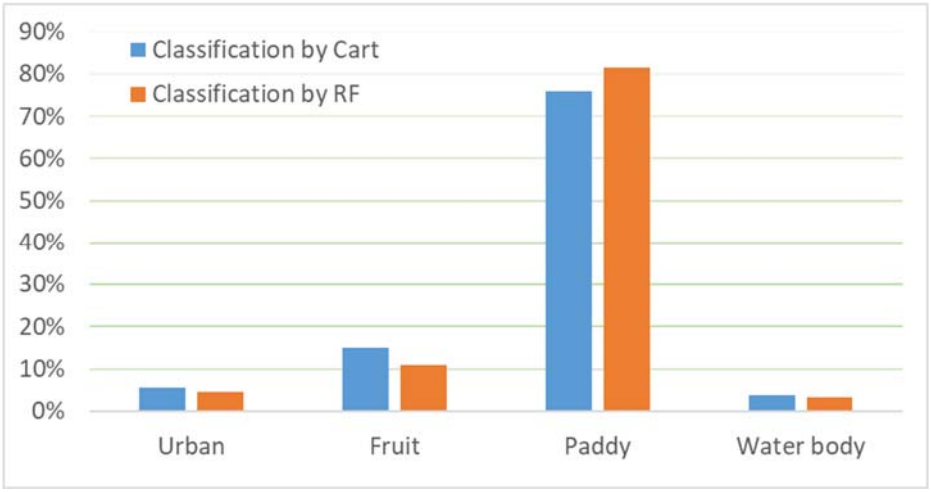
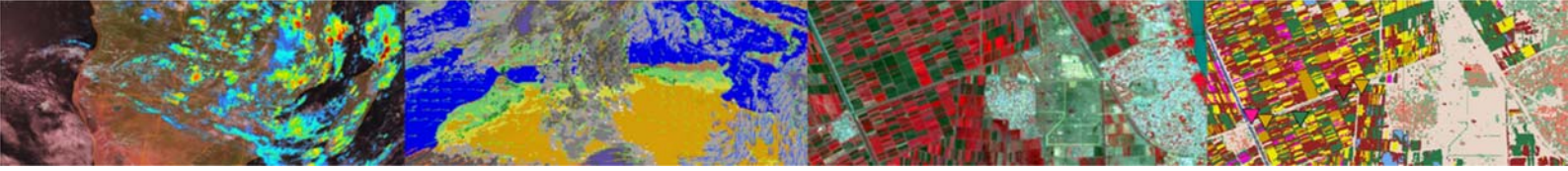


Figure 11 - Percentages of land cover types in dry season 2020

Conclusion

The monitoring land cover change play an important role in the environment and natural resources management that aims to achieve sustainable development in the Mekong Delta region. Due to the high spatial resolution, Sentinel-2 data can achieve high accuracies in land cover classification in Cantho city, Mekong Delta Vietnam. The classification results with random forest (RF) have higher accuracy compared to the Neural Network classifier, Classification And Regression Tree (CART).



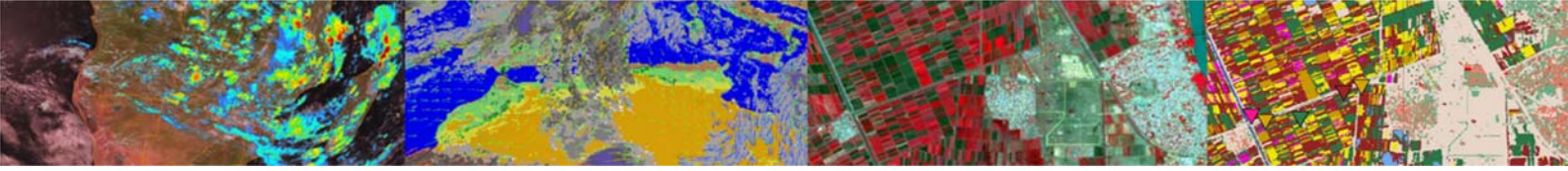
References

Jucker, T., J. Caspersen, J. Chave, C. Antin, N. Barbier, F. Bongers, M. Dalponte, K. Y. van Ewijk, D. I. Forrester and M. Haeni (2017). Allometric equations for integrating remote sensing imagery into forest monitoring programmes. Global change biology, 23(1), 177-190.

Li, X., D. Chen, Y. Duan, H. Ji, L. Zhang, Q. Chai and X. Hu (2020). Understanding Land use/Land cover dynamics and impacts of human activities in the Mekong Delta over the last 40 years. Global Ecology Conservation, 22, e00991

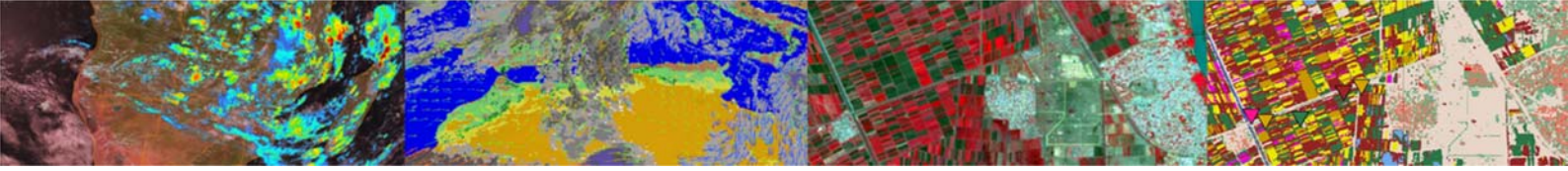
Malenovský, Z., H. Rott, J. Cihlar, M. E. Schaepman, G. García-Santos, R. Fernandes and M. Berger (2012). Sentinels for science: Potential of Sentinel-1,-2, and-3 missions for scientific observations of ocean, cryosphere, and land. Remote Sensing of Environment, 120, 91-101.

Phiri, D., M. Simwanda, S. Salekin, V. R. Nyirenda, Y. Murayama and M. Ranagalage (2020). Sentinel-2 data for land cover/use mapping: a review. Remote Sensing of Environment, 12(14), 2291.

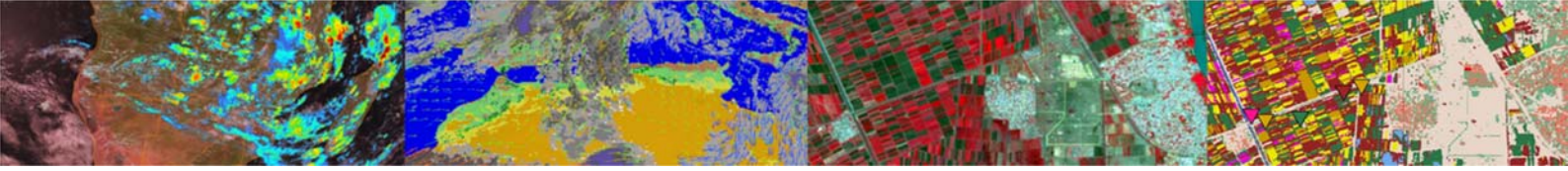


Annex 1: JAVA Scripts

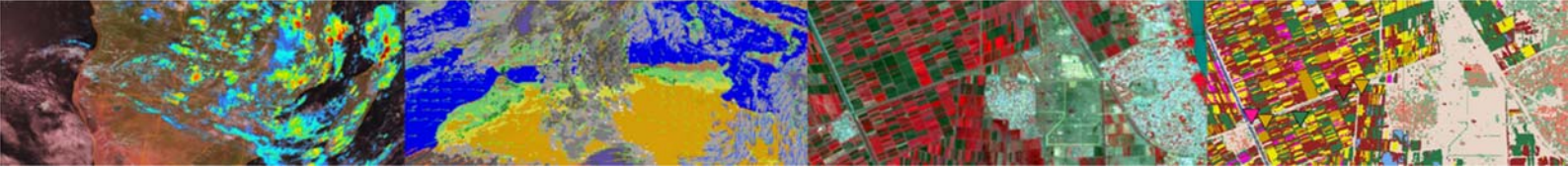
```
// Function to mask clouds using the Sentinel-2 QA band
//@param {ee.Image} image Sentinel-2 image
// @return {ee.Image} cloud masked Sentinel-2 image

function maskS2clouds(image) {
  var qa = image.select('QA60');

  // Bits 10 and 11 are clouds and cirrus, respectively.
  var cloudBitMask = 1 << 10;
  var cirrusBitMask = 1 << 11;

  // Both flags should be set to zero, indicating clear conditions.
  var mask = qa.bitwiseAnd(cloudBitMask).eq(0)
      .and(qa.bitwiseAnd(cirrusBitMask).eq(0));

  return image.updateMask(mask).divide(10000);
}

var data = ee.ImageCollection('COPERNICUS/S2')
                .filterDate('2020-03-01', '2020-06-01')
                .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE',10))
                .filterBounds(roi)
                .map(maskS2clouds)
                .median()

//Image visualization
var vis = {
  min: 0.0,
  max: 0.39,
  bands: ['B11','B8','B4']
};

var ndvi = data.expression('(NIR - Red) / (NIR + Red)', {
          'NIR':data.select('B8'),
          'Red':data.select('B4')
          })

var ndwi = data.expression('(Green - NIR) / (Green + NIR)', {
          'NIR':data.select('B8'),
          'Green':data.select('B3')
          })

var savi = data.expression('((NIR - Red) / (NIR + Red + 0.5)) * (1.0 + 0.5)', {
          'NIR':data.select('B8'),
          'Red':data.select('B4')
          })

//Add additional bands
var final_image = data.addBands(ndvi.rename('NDVI'))
                  .addBands(ndwi.rename('NDWI'))
                  .addBands(savi.rename('SAVI'))

// Bands for prediction
var bands = ['B2','B3','B4','B5','B8','B9','B11', 'NDVI', 'NDWI', 'SAVI']

//Print(final_image)
Map.addLayer(final_image.clip(roi), vis, 'Image')

var samples = urban.merge(waterbody).merge(paddy).merge(fruit)
```



```
var total_sample = final_image.select(bands).sampleRegions({
  collection: samples,
  properties: ['Property', 'landcover'],
  scale: 20
})
print(total_sample.size())

// Train a random forest classifier with 100 trees and others as default parameters.
var trainRf = ee.Classifier.smileRandomForest(10).train({
  features: total_sample,
  classProperty: 'landcover',
  inputProperties: bands
})

// Train a CART classifier with default parameters.
var trainCart = ee.Classifier.smileCart().train({
  features: total_sample,
  classProperty: 'landcover',
  inputProperties: bands
});


// Classify the image with the same bands used for training.
var classifiedCart = final_image.select(bands).classify(trainCart);
var classifiedRf = final_image.select(bands).classify(trainRf);

// Display the classification result and the input image

Map.addLayer(classifiedCart.clip(roi), {min: 0, max: 3,
      palette: ['red', 'green', 'yellow', 'blue']},
       'classifiedCart')

Map.addLayer(classifiedRf.clip(roi), {min: 0, max: 3,
      palette: ['red', 'green', 'yellow', 'blue']},
       'classifiedRf')

//ACCURACY ASSESSMENT

//Merge validation polygons into one feature collection

var valsamples = vurban.merge(vwaterbody).merge(vpaddy).merge(vfruit);

var validation = final_image.sampleRegions({
  collection: valsamples,
  properties: ['landcover'],
  scale: 20
});
print(validation);

// Classify the validation data.
var valCart = validation.classify(trainCart);
var valRf = validation.classify(trainRf)

//Compare the land cover of the validation data against the classification result

var testAcCart = valCart.errorMatrix('landcover', 'classification');

var testAcRf = valRf.errorMatrix('landcover', 'classification');
```



```
//Print the error matrix to the console
print('##### TRAINING ACCURACY #####');
print('Cart: Validation error matrix: ', testAcCart);
print('RF: Validation error matrix: ', testAcRf);
print('Cart: Validation overall accuracy: ', testAcCart.accuracy());
print('Rf: Validation overall accuracy: ', testAcRf.accuracy());

//Export the classification result
Export.image.toDrive({
  image: classifiedCart,
  description: "classCart_dry2020",
  folder: "GEE",
  scale: 10,
  region: roi,
  maxPixels: 1e13
});

Export.image.toDrive({
  image: classifiedRf,
  description: 'classRf_dry2020',
  folder: "GEE",
  scale: 10,
  region: roi,
  maxPixels: 1e13
});

//End
```



10. Water quality monitoring in Red River downstream using remote sensing data and spatial regression method

Trinh Thi Tham (tttham@hunre.edu.vn)

Introduction

The Red River is the largest river in northern Vietnam. The river originates from Yunnan Province (China) and flows through several provinces and cities in Vietnam such as Lao Cai, Yen Bai, Phu Tho, Vinh Phuc, Hanoi, Hung Yen, Ha Nam, Nam Dinh, and Thai Binh before emptying into the East Sea. The river is 1150 km long with 800 km of meander through steep mountains and highlands where violent and unpredictably swell during the rainy season. When the river reaches the Red River Delta, at an elevation above sea level of about three meters, it turns quiet and gentle. The Red River Delta plays an important role in socio-economic development with a variety of historical and cultural values. In recent years, many extensive industrial parks and key economic regions have been built in this delta area. Improper emissions from industrial parks, livestock farms, and domestic sectors have caused significant impacts on water quality of the Red River Delta. The 2018 Annual Report of the Ministry of Natural Resources and Environment showed that water quality of the Red River has tended downwards over the last 30 years (MONRE 2018). The water and sediment contamination with organic and inorganic pollutants in the delta has been documented by a number of previous studies. For example, arsenic concentration in the water of the Red River in Hanoi was 10 µg/L, which exceeded the WHO provisional guideline value (Berg et al. 2001). The average concentrations of some heavy metals (V, Zn, Cu, Cr, Pb, Ni and Zn) in sediments collected from Red River ranged from 0.35 to 836 ng/g dry weight (Nguyen, T.T.H, et al. 2016). (Kiểm tra lại thông tin, xem xét bổ sung thông tin về các chỉ tiêu cơ bản và chất hữu cơ).

Objective

- Use sentinel-2 images to determine instantaneous minimal variation of water quality parameters.
- Comparison of the results obtained by the Sentinel-2 sensors with actually measured data in real water samples

Method and software tools selected

Method and results

Using Sentinel 2B

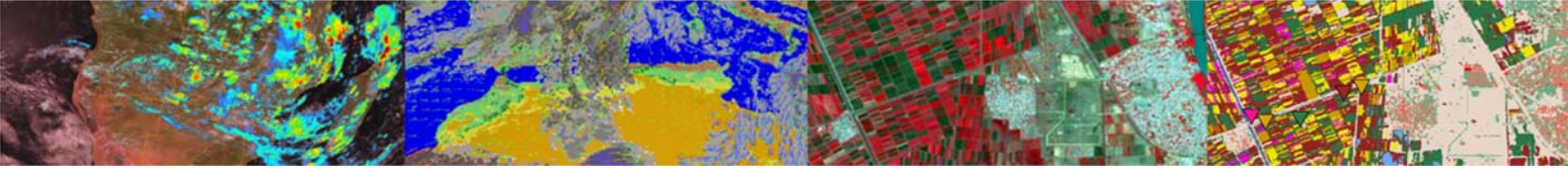
The data were downloaded from https://scihub.copernicus.eu/dhus/#/home

Establishing regression equation using Sentinel 2B level 1

The in-situ measurement was conducted in 22-23 February and 06-07 June 2021; however, I can only download the Sentinel 2B at 07 February and 17 June 2021.
The data downloaded is level 1: without atmospheric correction and the values are in radiances. The file names are:

- S2B_MSIL1C_20210207T032919_N0209_R018_T48QXH_20210207T060355.SAFE
- S2B_MSIL1C_20210617T032539_N0300_R018_T48QXH_20210617T061037.SAFE

First, the image is atmospherically corrected using Preprocessing option in the Semi-Automatic Classification Plugin in QGIS as in the following figure:



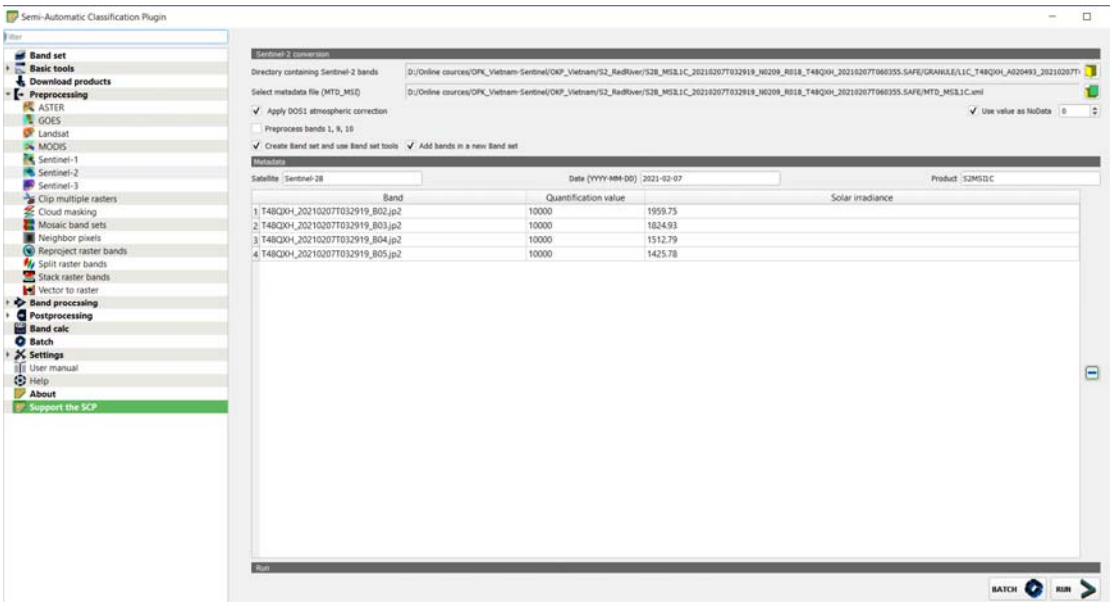


Figure 1. Semi-Automatic Classification Plugin

Then, the corrected image is converted from radiance to reflectance by dividing by 10,000 using Raster Calculator in QGIS.
A polygon shapefile of the study area is plotted into the map to mask the big Sentinel 2 image to then smaller shapefile only where the measurement was conducted. Using Raster Extraction to mask the map.
Then, I use Tool “Add delimited text layer” to add sampling points on the map as:

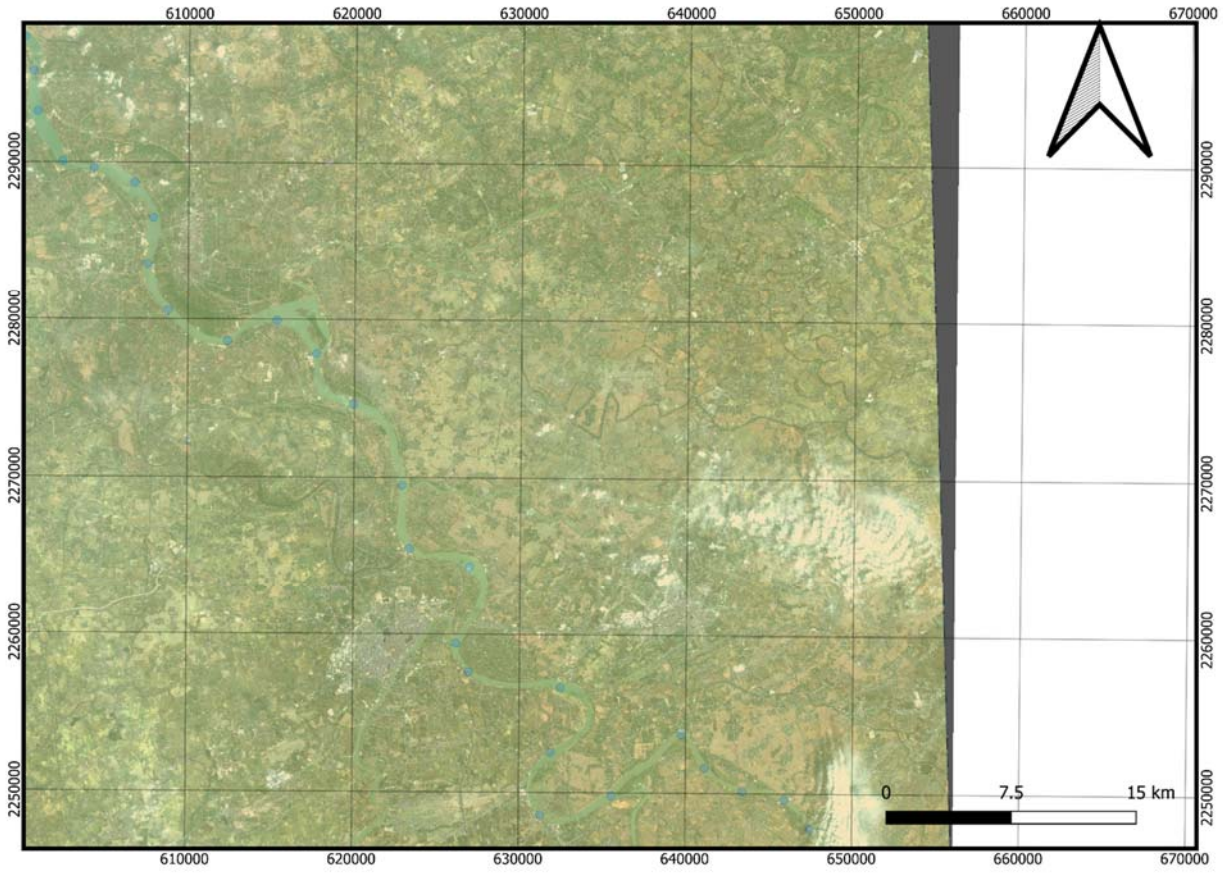
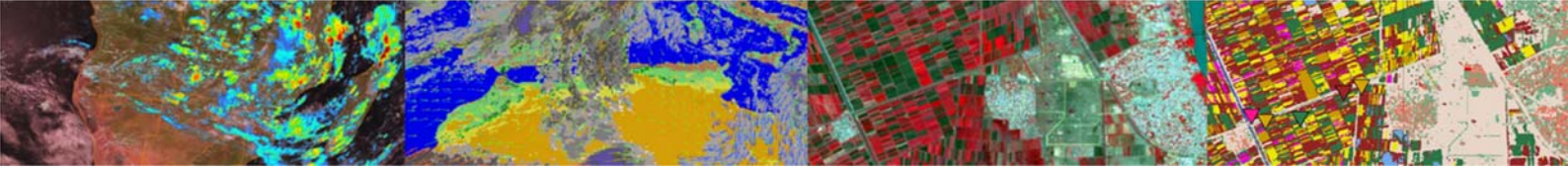
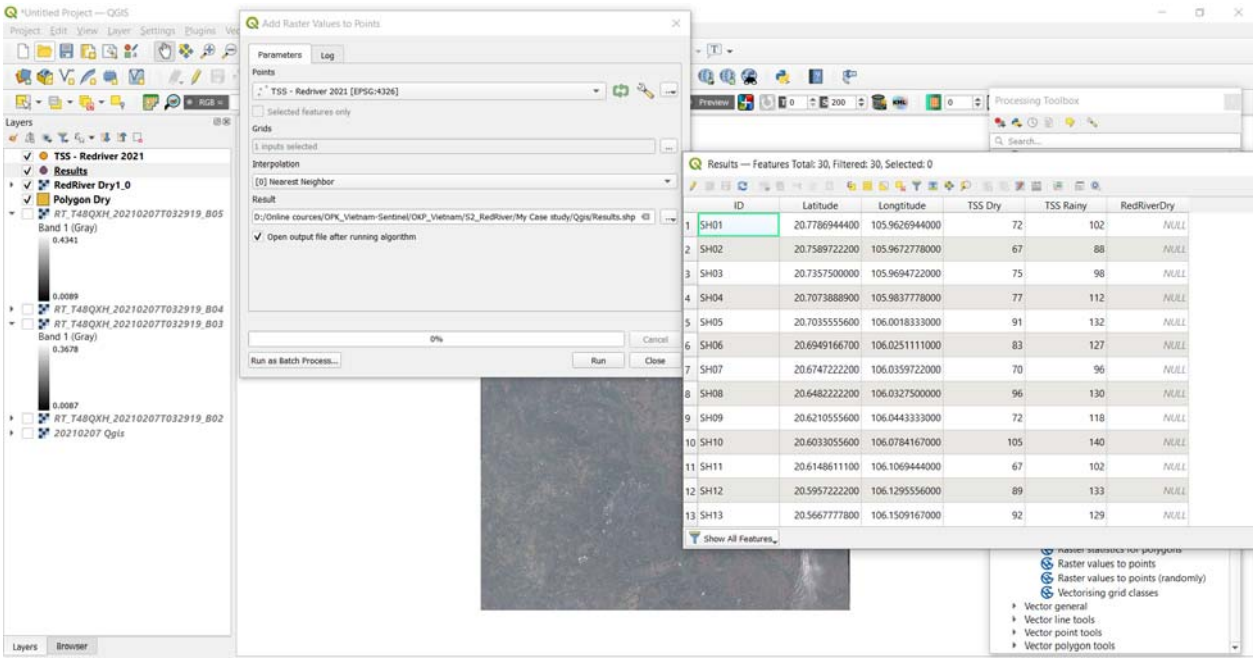


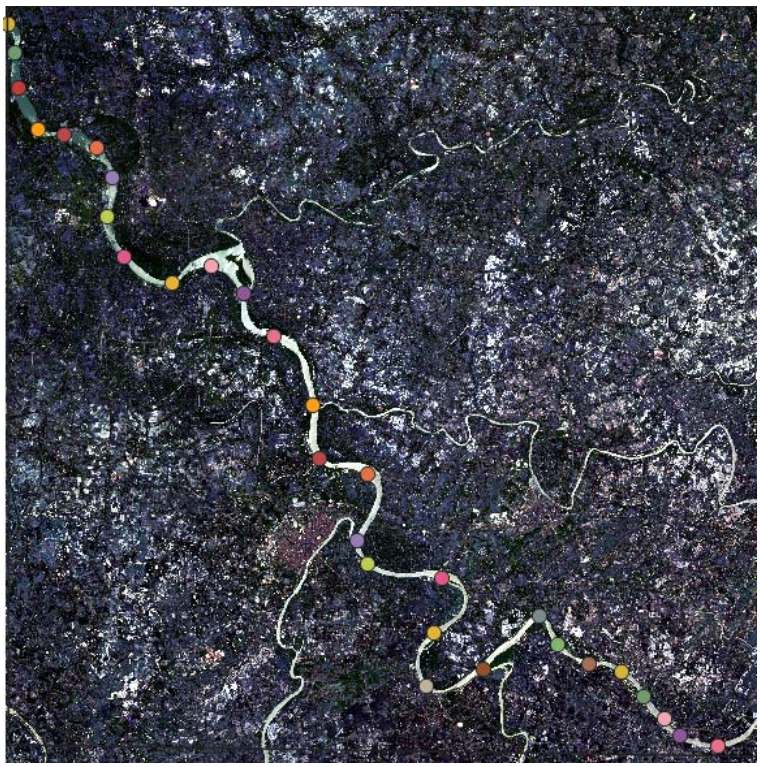
Figure 2: Research area map



However, I see the big mistake when I extract raster to point as shown in the picture below:

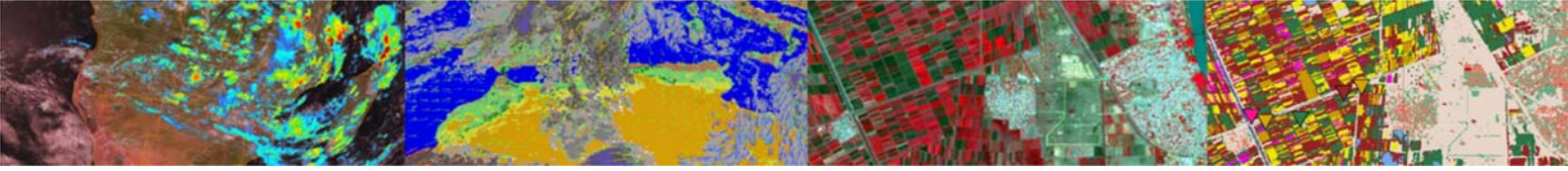


For Sentinel image collected on 17 June 221, I have the same results. I think, there are some problem with my knowledge about Qgis.



I am trying to fix this problem.

Acknowledgement
I am sincerely thankful to Juliette Eulderink and Prof. Ben Maathuis for organizing the course. Thanks for useful lessons from Prof. Ben.



11. GIS Methodology applying for managing Eutrophication in the West lake in Hanoi, Vietnam

Phạm Thị Hong (hongpt@tlu.edu.vn)

Introduction

Hanoi city was built up on the wetland land of the Red River delta therefore it has a network of more than 200 lakes. Most of them are eutrophic lakes and degradation in water quality and aquatic biodiversity is common in most of these urban lakes.

The development of urbanization in Hanoi are causing a large amount of nutrient discharged into the Hanoi lake, result in increasing the eutrophication of water bodies (Bhagowati, Talukdar, and Ahamad 2020). Eutrophication management strategies have been related to nutrient reduction which might depend on the rainfall, waste water runoff, and sediment. Other factors would affect to the eutrophication statement are temperature and water level. Previous decades, managing eutrophication in water body need a lot effort to monitor water quality by equipment and sensors to measure Water quality indicators, such as chlorophyll-a (Chl-a), total suspended matter, turbidity, depth of the secchi disc, and colored dissolved organic. However this approach has time and space limitation. Satellite-borne spectrometric sensors are capable of detecting phytoplankton growth and composition (Sòria-Perpinyà et al. 2020). In the past, most of research on monitoring water quality were based on remote sensing employed satellite data from MERIS (Medium Resolution Imaging Spectrometer) (Hu et al. 2004) and MODIS (Moderate Resolution).

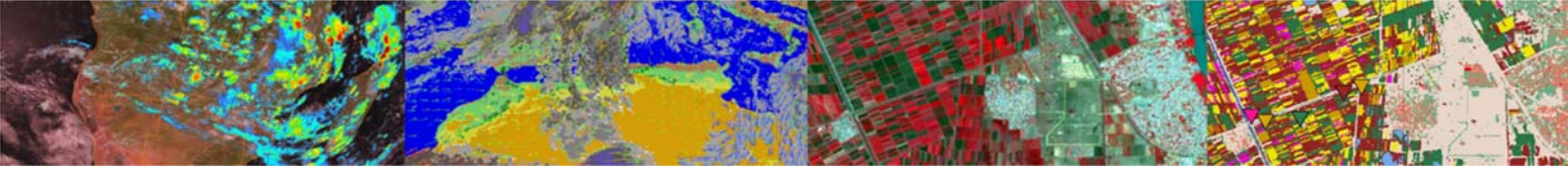
Imaging Spectroradiometer)(Carpenter and Carpenter 1983), and Landsat (Zhao et al. 2020). However, due to the limitation of spatial resolution and longtime observation, few of studies were conducted. The new series of Copernicus satellites from the European Union's Earth Observation Program called Sentinel have now brought the Sentinel-2 (S2-A and B) into service with an MSI (Multispectral Instrument) sensor. The imagery features a spatial resolution of 10 m, 20 m, and 60 m, meaning that even small lakes can be studied and monitoring of water color anomalies is possible offering future continuous observation possibilities at high spatial resolution.

West Lake is the biggest and important lake in Hanoi. The main function of this lake is regulating the rainfall and maintaining the biodiversity of the water body in Hanoi Lake. However, these years, due to the discharging of untreated domestic waste water from surrounding and over effluences, the increasing of eutrophication cause the decreasing of biodiversity. In this study, a spatial approach is presented to assess the trophic states of the West lake based on the geographical information system (GIS) with spatiotemporal attributes to have better eutrophication management strategies.

Materials and Methods

The dataset used in this study is the standard Sentinel-2 Level-1C product, produced by radiometric and geometric corrections, provides spatial registration on a global reference system with sub-pixel accuracy. The Sentinel-2 Level-1C product comprises 100 km ×100 km tiles in the UTM/WGS84 projection and provides the Top-Of-Atmosphere (TOA) reflectance.

The Sentinel-2 Level-1C images were downloaded from the ESA Sentinel-2 Pre Operations Hub (https://scihub.copernicus.eu/ (accessed on 23 August 2021). All satellite data processing was performed using the Sentinel Application Platform (SNAP). Sentinel-2 is composed of bands with a spatial resolution of 10 m (band 3, band 4, and band 8) In this case, bands 3 (560 nm), 4(665 nm) and 5 (705 nm) were analyzed. These bands were chosen because the



reflectance peak between 700 and 720 nm has been used for estimating the Chl-a concentration in lake waters.

The images were downloaded for two periods in June and July of 2021. The spectral indices used were the Normalized Difference Water Index (NDWI), the Normalized Differences Vegetation Index (NDVI), the green Normalized Difference Vegetation Index (gNDVI).

Results and Discussion

The Fig 1 shows the natural color image of the West Lake in Hanoi. However, the image has cloud cover and difficult to see clearly.

Fig 1. Natural color composite image of the West lake in Hanoi

The NDVI index was calculated to inspire the change of chlorophyll content in the two period from June to July. However, in the Fig 2, we can see the change in color of river but not in the lake. The result would be that this NDVI index is not very sensitive to change in chlorophyll content, which is related to changes in nitrogen content in reservoirs. Even there was significant change in the color of the Red River, there was no change in the color of the West lake. These indicate that the eutrophication of the West Lake does not affect by the Red river.

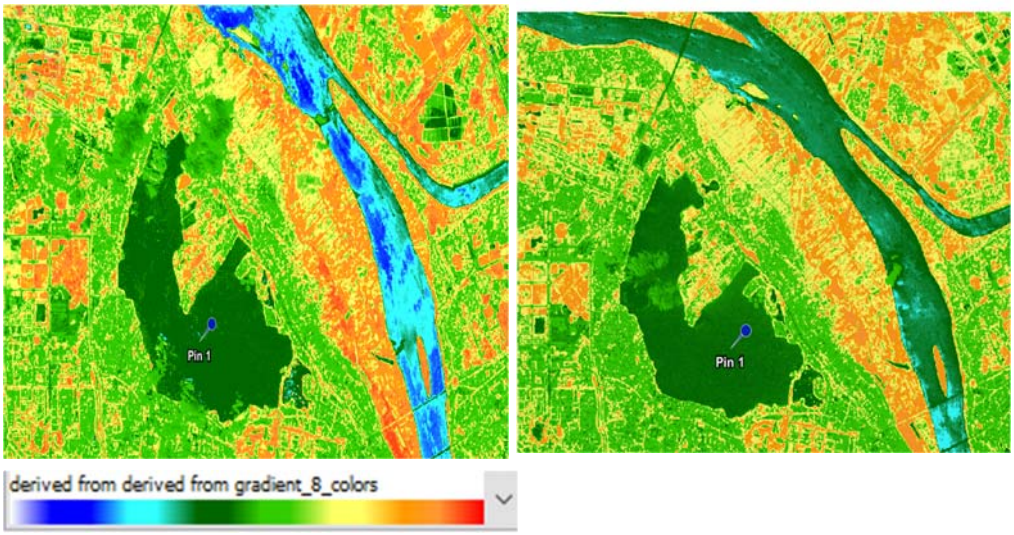
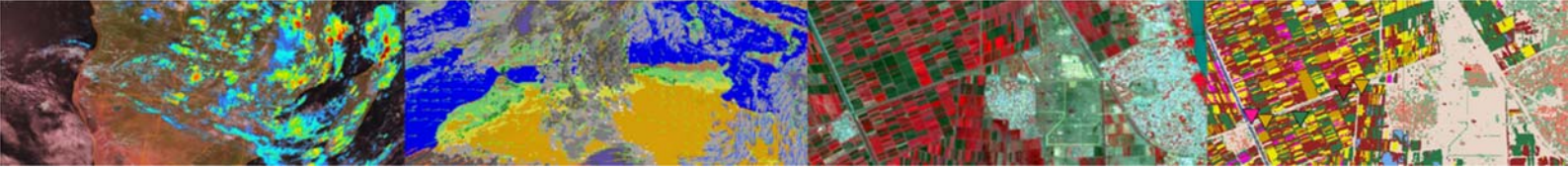


Fig 2 a) NDVI in June 2021 ***b) NDVI in July 2021***



However the sensitivity could effect by different factors especially the composition of the algae in the lake and the optical depth of water. It is difficult to estimate the properties of the water column when the remote sensing signal comes mainly from the bottom of the lake or when those properties vary significantly over the course of the water column.

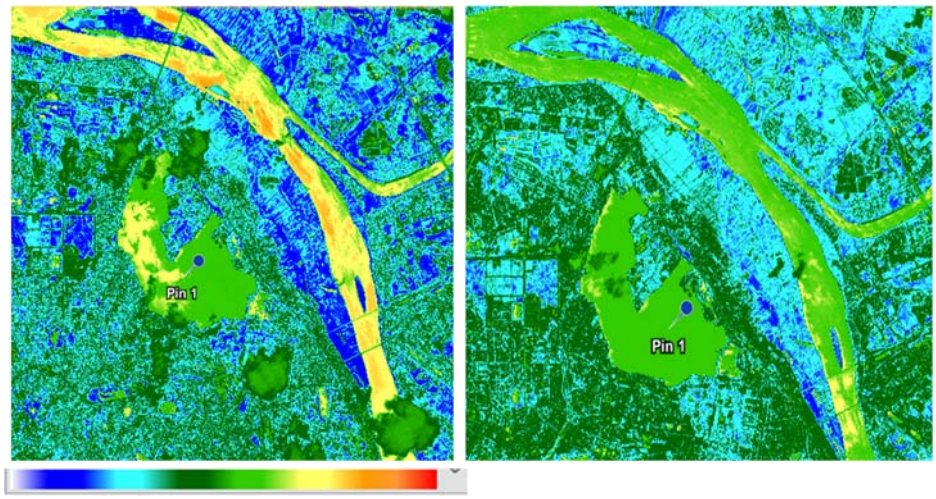


Fig 3: a) NDWI in June 2021 b) NDWI in July in 2021

However in the same period, with the NDWI index see the significant change in the color in the two periods of the West lake. In June the color near embankment is turned into yellow color indicating the higher concentrations of chlorophyll-A. While in July the color in this location changed into green. The change of color in the two images indicate that, the waste water discharge into the lake may cause the development of algae in the lakes.

Conclusions

The use of satellite images as a tool for analyzing cyanobacterial blooms is an innovative technology that will facilitate water governance and help develop measures to guarantee water security. Different bands of the Sentinel-2 satellite are analyzed to find the reason of eutrophication statement of the West Lake. However to better evaluation of applicability of this method, the result of the analysis with Sentinel 2 should be compared with the field data to check the correlation.

References

Bhagowati, Biswajit, Bishal Talukdar, and Kamal Uddin Ahamad. 2020. 'Lake Eutrophication: Causes, Concerns and Remedial Measures.' in, *Emerging Issues in the Water Environment during Anthropocene* (Springer).

Carpenter, DJ, and SM Carpenter. 1983. 'Modeling inland water quality using Landsat data', *Remote Sensing of Environment*, 13: 345-52.

Hu, Chuanmin, Zhiqiang Chen, Tonya D Clayton, Peter Swarzenski, John C Brock, and Frank E Muller–Karger. 2004. 'Assessment of estuarine water-quality indicators using MODIS medium-resolution bands: Initial results from Tampa Bay, FL', *Remote Sensing of Environment*, 93: 423-41.

Sòria-Perpinyà, Xavier, Eduardo Vicente, Patricia Urrego, Marcela Pereira-Sandoval, Antonio Ruíz-Verdú, Jesús Delegido, Juan Miguel Soria, and José Moreno. 2020. 'Remote sensing of cyanobacterial blooms in a hypertrophic lagoon (Albufera of València, Eastern Iberian Peninsula) using multitemporal Sentinel-2 images', *Science of the Total Environment*, 698: 134305.

Zhao, Yelong, Qian Shen, Qian Wang, Fan Yang, Shenglei Wang, Junsheng Li, Fangfang Zhang, and Yue Yao. 2020. 'Recognition of water colour anomaly by using Hue Angle and Sentinel 2 image', *Remote Sensing*, 12: 716.