

Manual for Forest and Land-use Classification

2023

Korea University

I. QGIS Overview

I-1. Introduction of QGIS Application

QGIS is a free and open-source desktop GIS application that allows users to create, edit, visualize, analyze and publish geospatial information. It has a user-friendly graphical user interface (GUI) and it provides various analysis tools. QGIS consists of four products that are QGIS Desktop, QGIS Browser, QGIS Server and QGIS Client. Among those products, QGIS Desktop is a basic program to process data and the others are usually used for searching or sharing geospatial data. Therefore, we will explain how to use QGIS Desktop as a supportive tool for decision making.

STEP 0. Install Q-GIS Application

1. Access to the site (<https://www.qgis.org/en/site/forusers/download.html>), and download the file.
2. Click **QGIS Standalone installer Version 3.28 (32/64bit)**

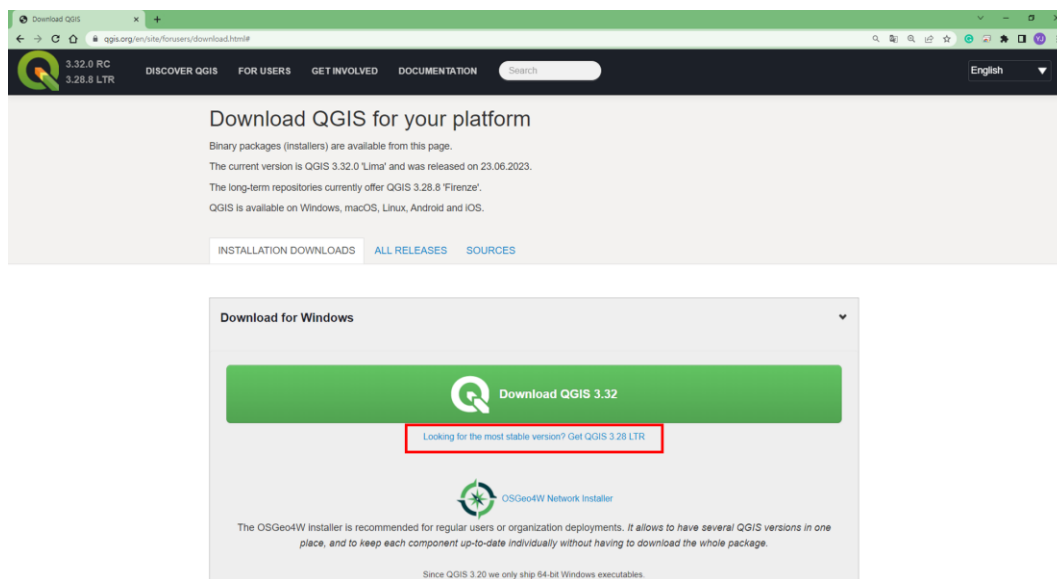


Figure 1 Selection of the QGIS software

3. Open Source Programme Before starting QGIS Programme, we can set option (language setting, toolbox setting, and etc.).
4. Setting language setting

1) Click Setting > Options > Override system locale > American English and C Default (C)

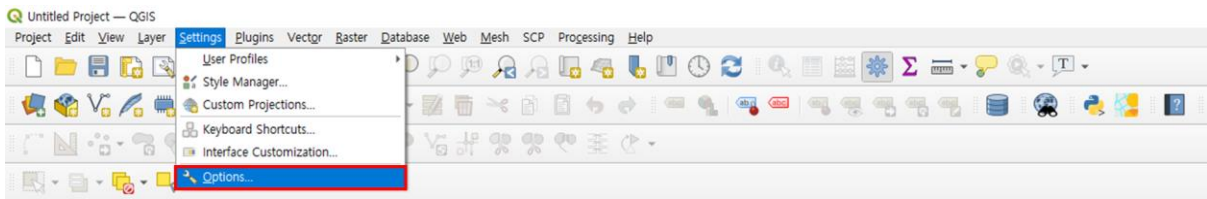


Figure 2 Setting and options button to change the language

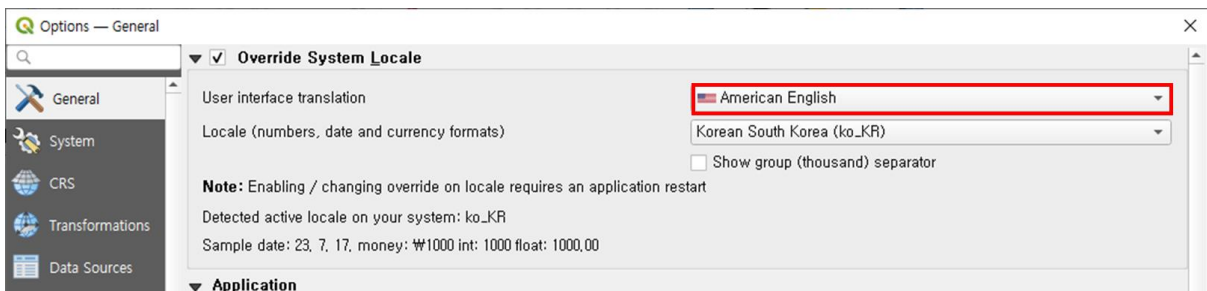


Figure 3 Language setting option

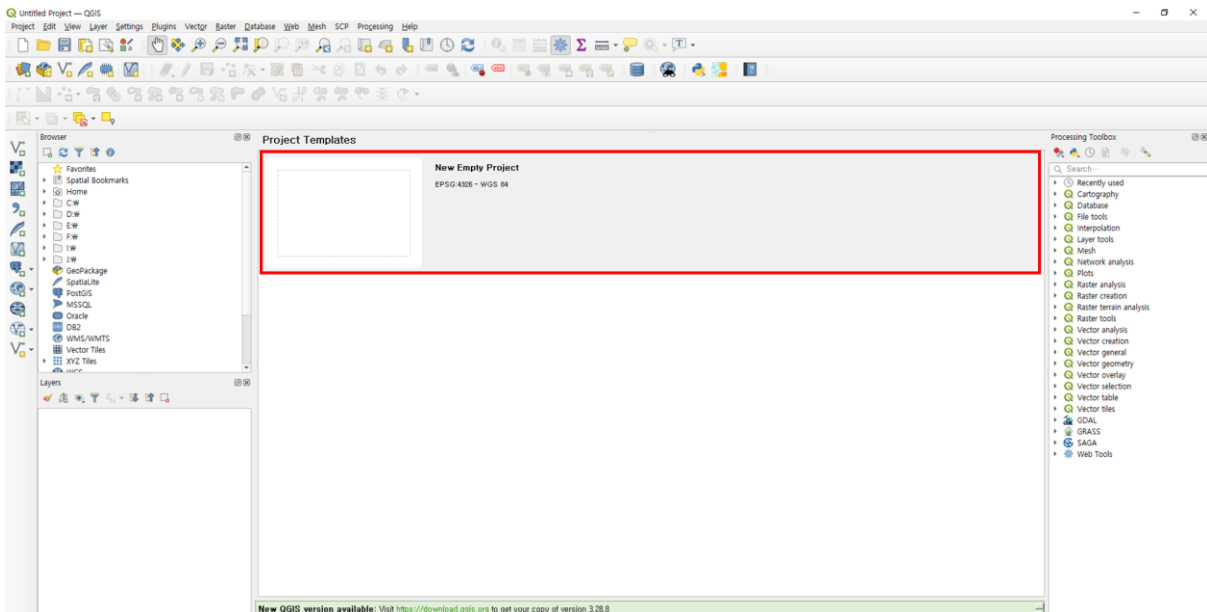


Figure 4 QGIS interface after changing language

When you open QGIS, you will have a new, blank map as a below picture (Figure 5). The main menu toolbar is placing at the upper side of the window and the toolbox is on the right side. Browser and layer windows are on the left side, and python console is on the lower side. Before starting any process through this program, you can change the default setting for your

convenience. For example, you can also change the default language of the program according to your country. On the main menu toolbar, there is a setting section and you will see options under that section. This will make it possible to change any options you want.

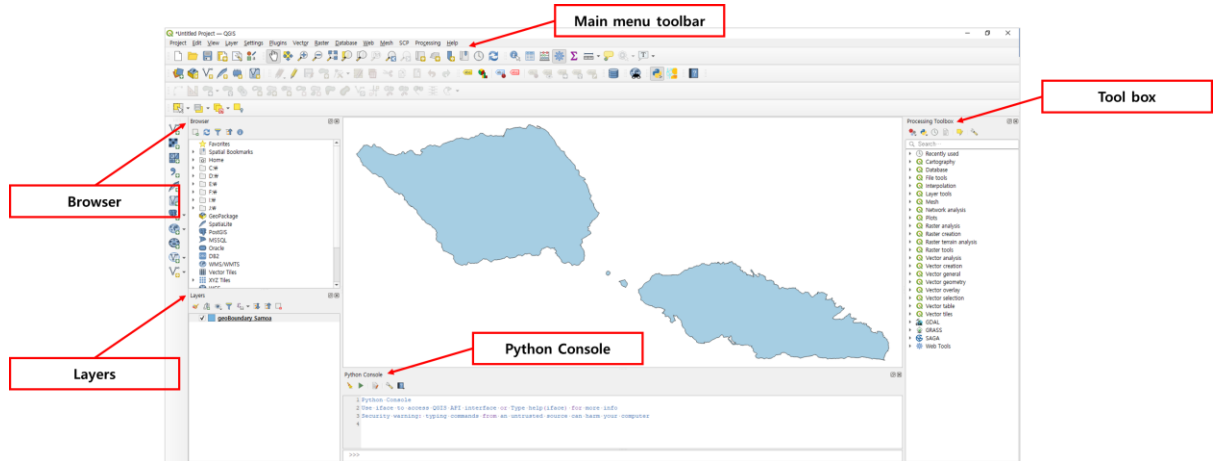


Figure 5 Introduce of QGIS interface

After every setting has done, you need to click the 'Open Data Source Manager' button to load a new file on this blank map and find the file that you want to use. The next step is processing data. This can be conducted on the 'Data Source Manager' window (Figure 6). It will come out by clicking the button on the left upper side and you will see the various method of data processing.

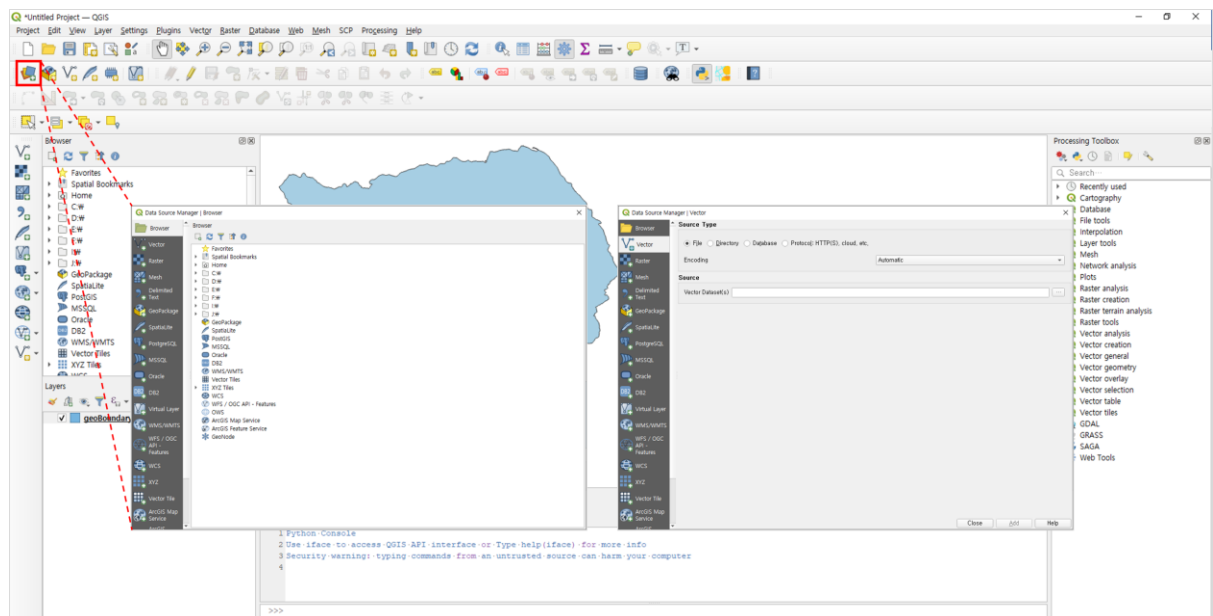


Figure 6 To conduct on a data source manager

STEP 1. Exploring QGIS

1. To rearrange the toolbar, select **View > Toolbars** to add extra toolbar

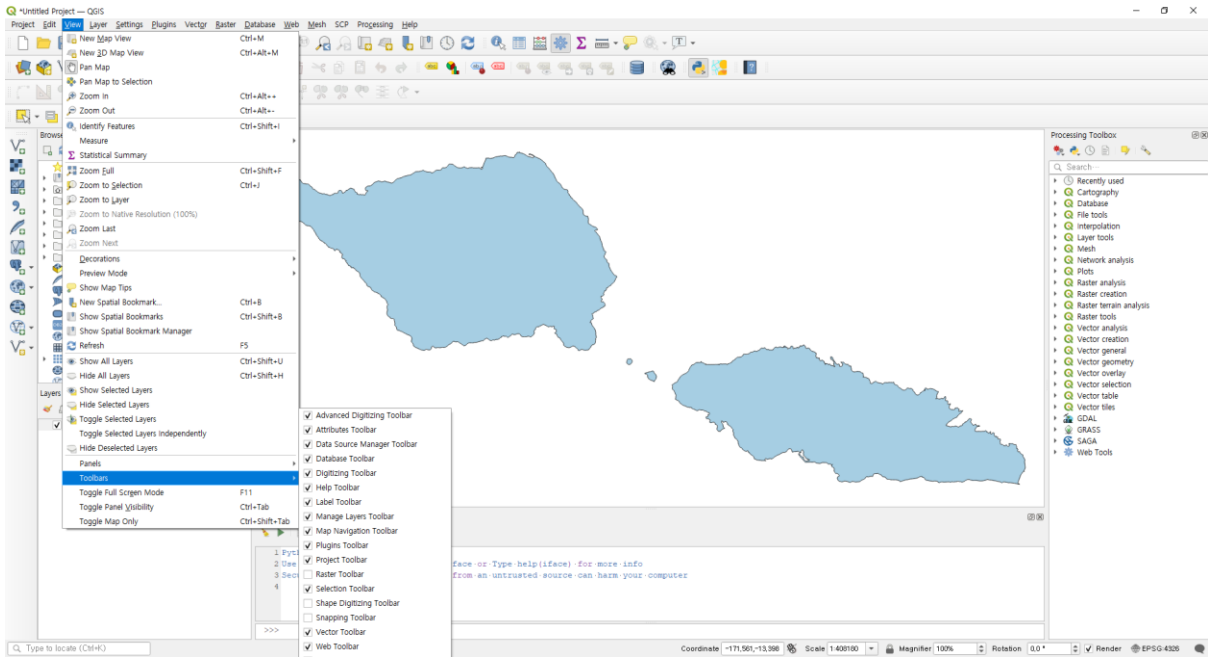


Figure 7 To rearrange the toolbar

2. To add data select **Layer > Add Layer > Add Vector / Raster Layer** or click **Data source manager**

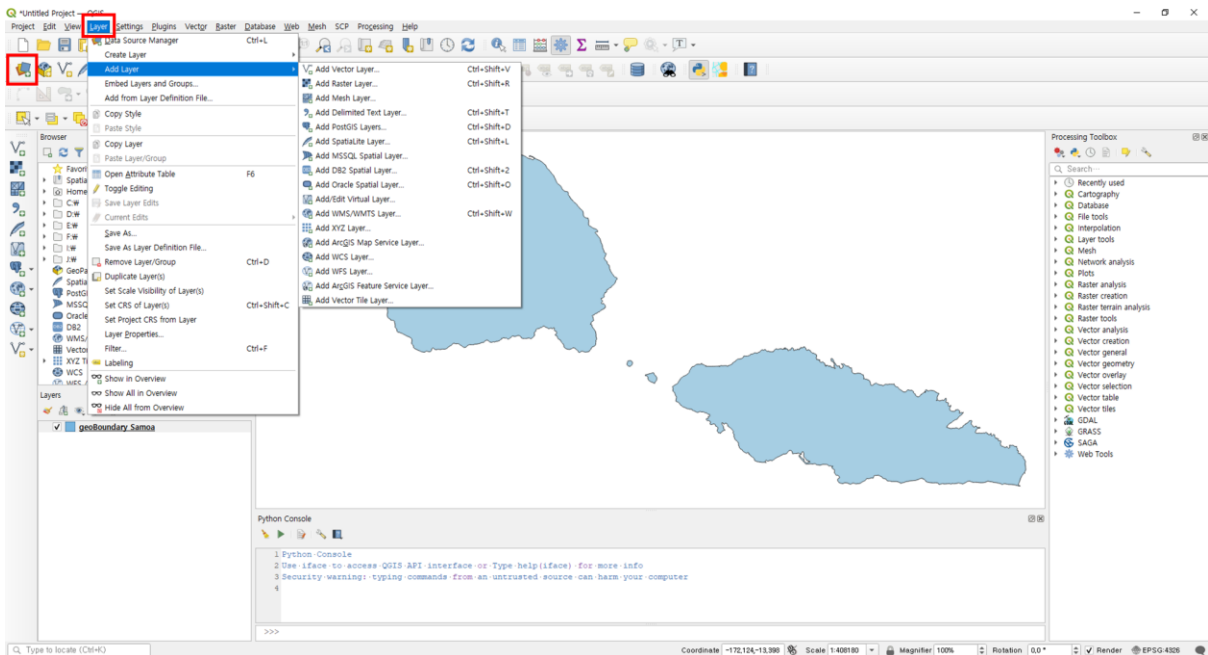


Figure 8 To add data

3. To view attribute data (Figure 9), click right point at the Layers Window on the left, click

on the name of data and select 'Open Attribute Table'

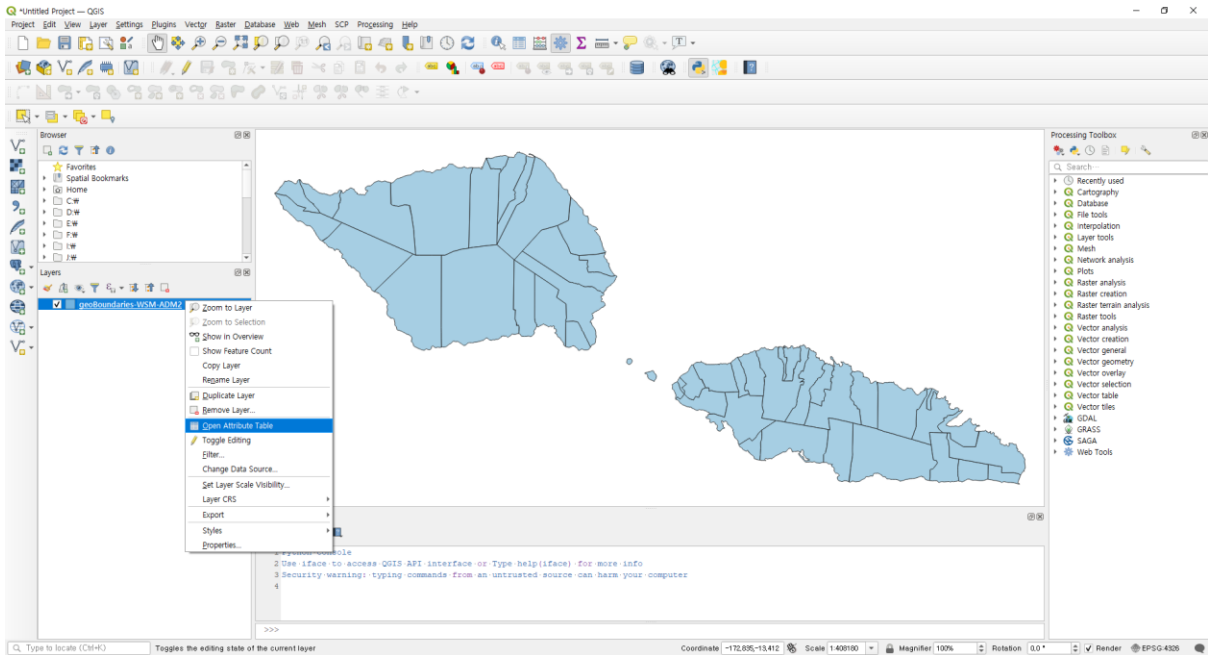


Figure 9 To view attribute data

- To overview layer (Figure 10), click View > Panels > Check 'Overview Panel' > Right click of mouse on the name of data and check 'Show in Overview'

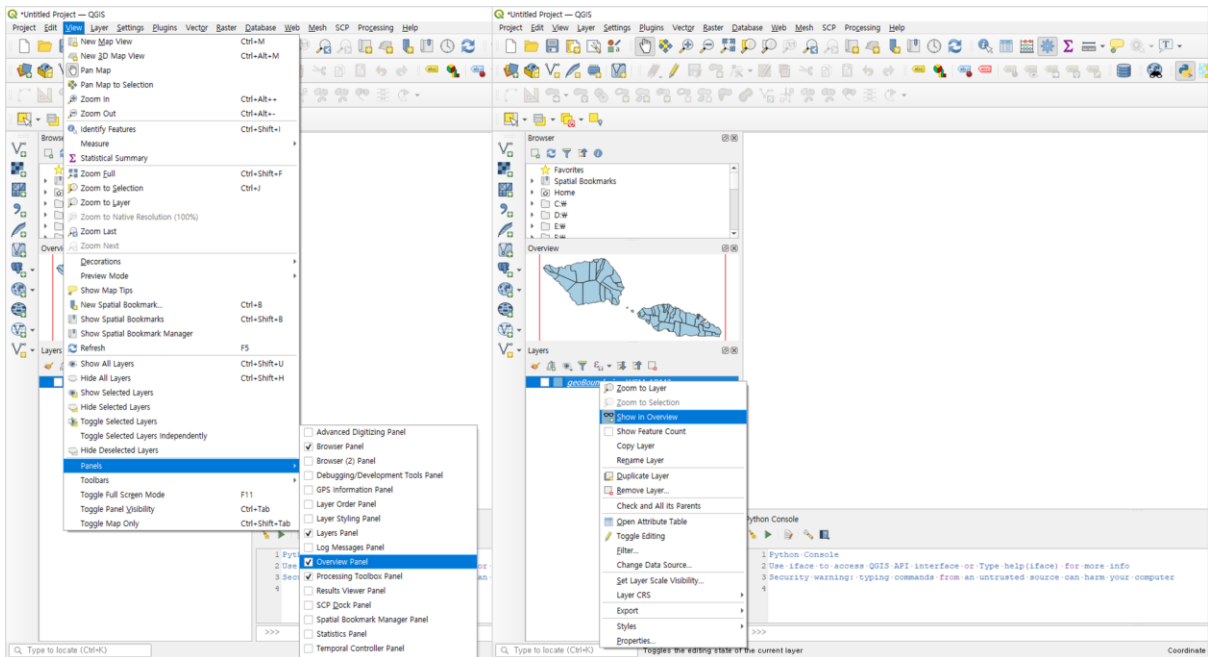


Figure 10 To overview layer

- To select the features (Figure 11), use toolbars (in the Red box) of attribute table

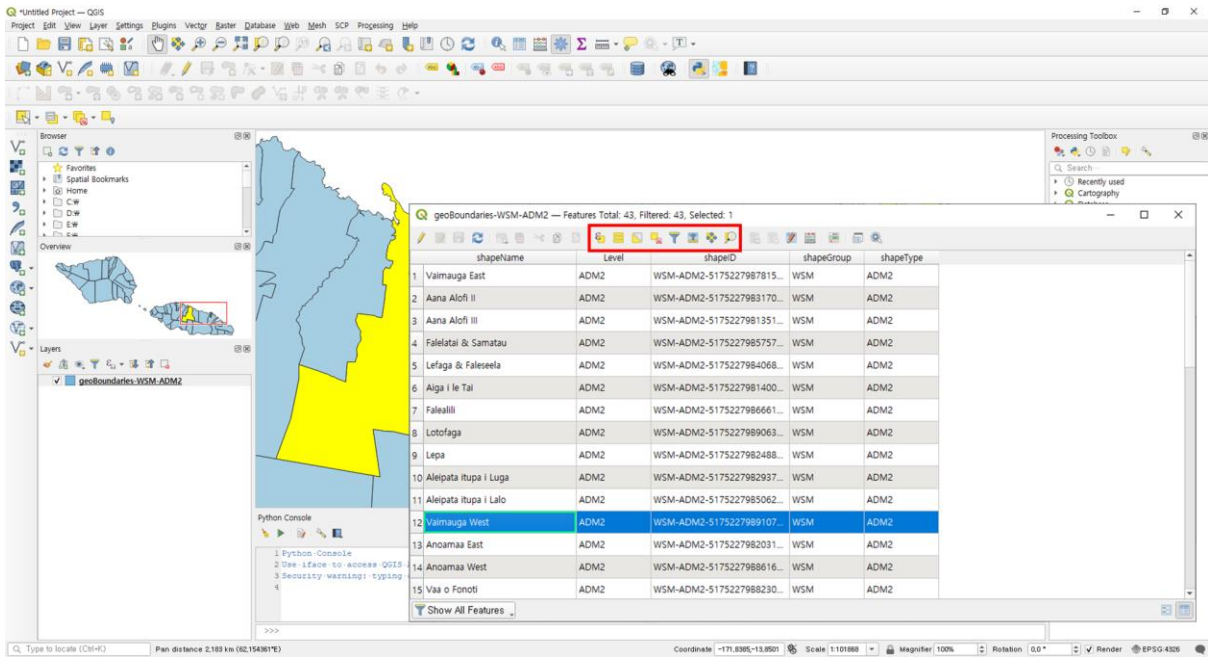


Figure 11 Selected feature

6. To search by using expression-based filter (Figure 12), open Attribute Table > Show All Features > Advanced Filter (Expression) > Choose one on the Function List

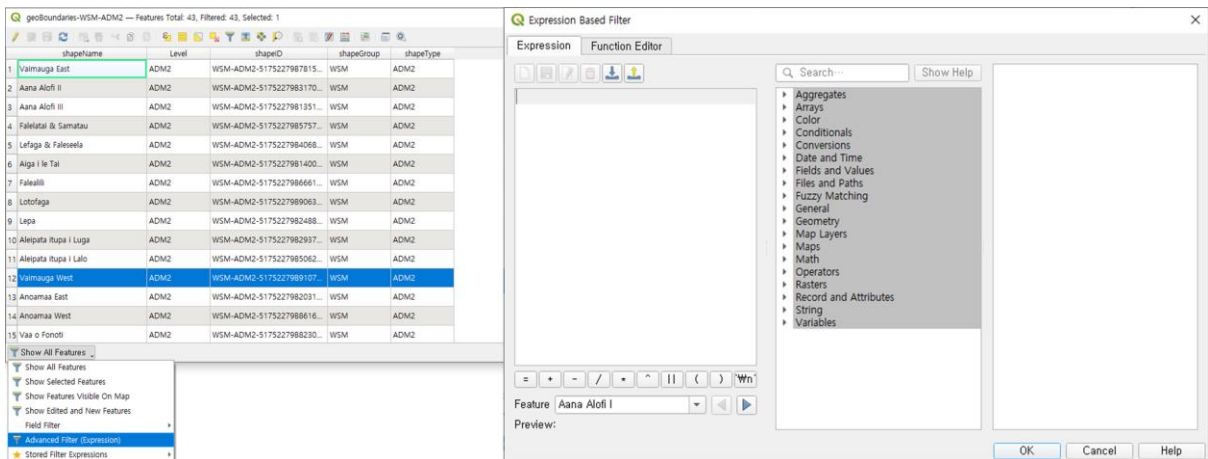


Figure 12 To search by using expression-based filter

7. To Create a new field, click (on the attribute table) **Toggle editing mode > New column**

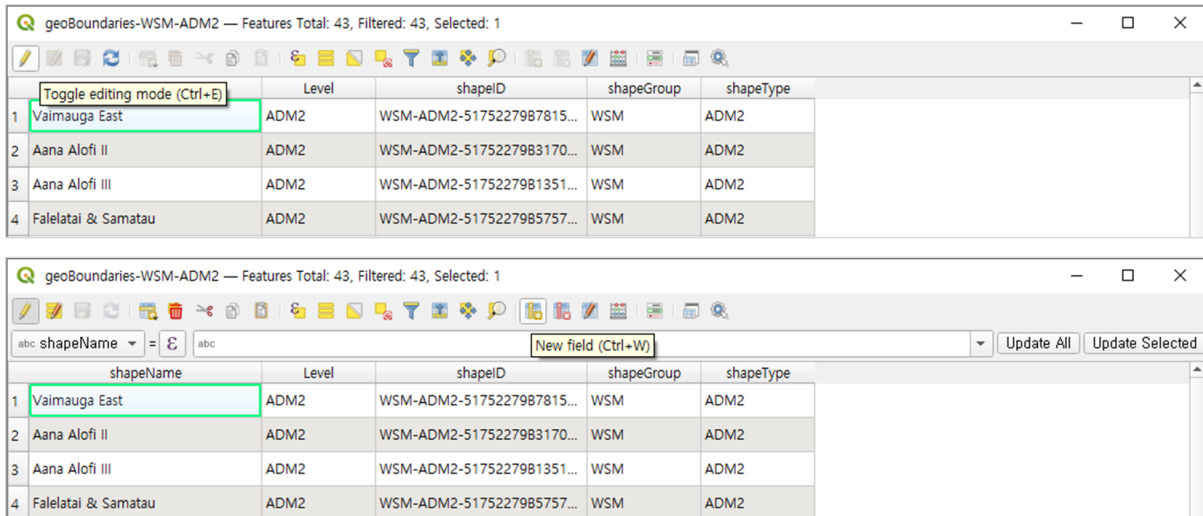


Figure 13 To create a new field

8. To open Layers Plugin (Figure 14), click **Plugins > Manage and Install Plugins > All > OpenLayers Plugin**

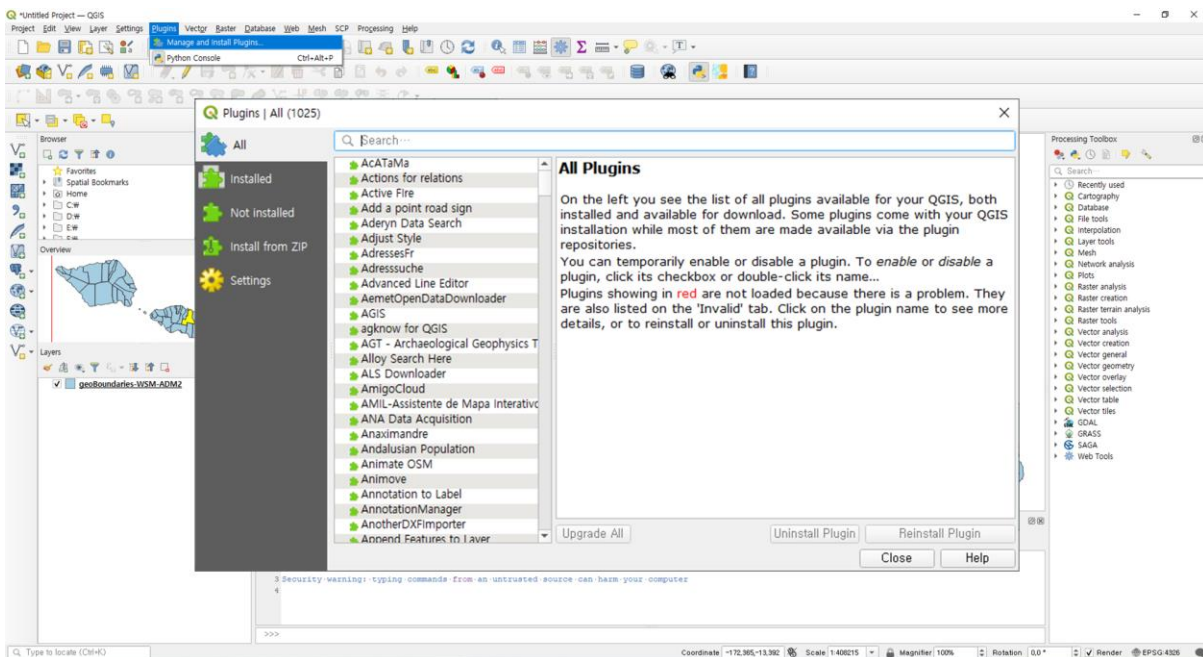


Figure 14 To open layers plugin

I-2. Data format

Before processing data, there are some important things in GIS to be noticed (Figure 15). In GIS, there are two data formats which are spatial data and non-spatial data. Regarding non-spatial data, it is statistical data that is presented as numbers or words. On the other hand, spatial data has a specific feature that presented as a shape. In addition, this shape can be made by different components. According to the components, Spatial data can be divided into two groups. One is vector and the other is raster.

Vector data consist of three types of geometry according to the geographical features which are point, line, and polygon. The geometry structures hold information in themselves and each geometry feature can carry multiple attributes instead of just one, as an example, a database of cities can have attributes for name, country, population, etc. In addition, data storage can be very efficient compared to raster data.

Raster data is useful for storing data that varies continuously, as in an aerial photograph, a satellite image or an elevation surface. Additionally, Raster data stores the data in the type of digital image represented by reducible and enlargeable grids and these grids of cells contains a value representing information, such as temperature, discrete data represents features such as land-use or soils data.

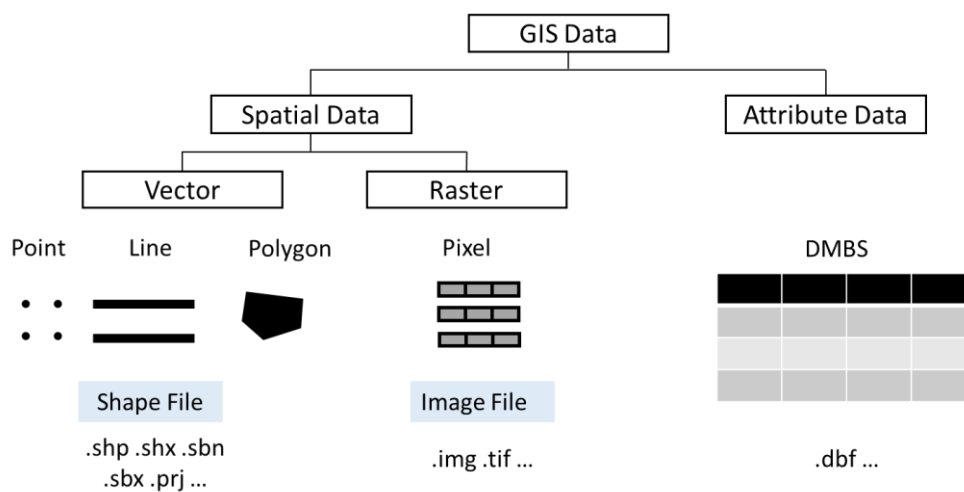


Figure 15 Various data format of GIS: vector, raster, and attribute data

I-3. Basic Tool (Map composition)

STEP 1. Add CSV File as Layer

1. Click **Layer > Add Layer > Add delimited text layer**
2. Open **Protected_Area_Samoa (... .csv)** dataset from source **(./KU_material/1_background/spi_table.txt)**
3. Set **Geometry Definition > Point coordinates** X, Y filed as long., lat.

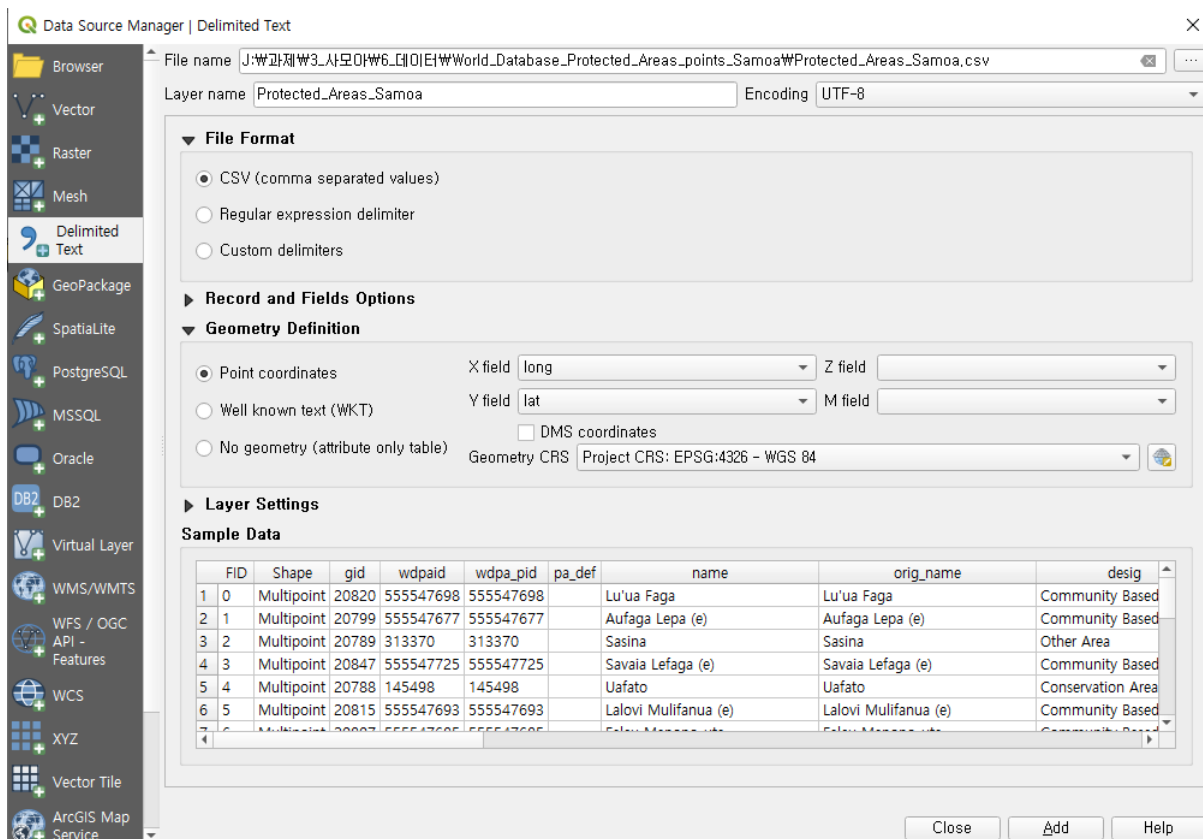


Figure 16 How to add CSV file



Figure 17 Protected area point data of Samoa

STEP 2. Save Layer as Point Shape File (Figure 18)

1. Right click on the layer
2. **Export** > **Save Features as** > select **ESRI (... shp)** as a Format > Input File name > Save

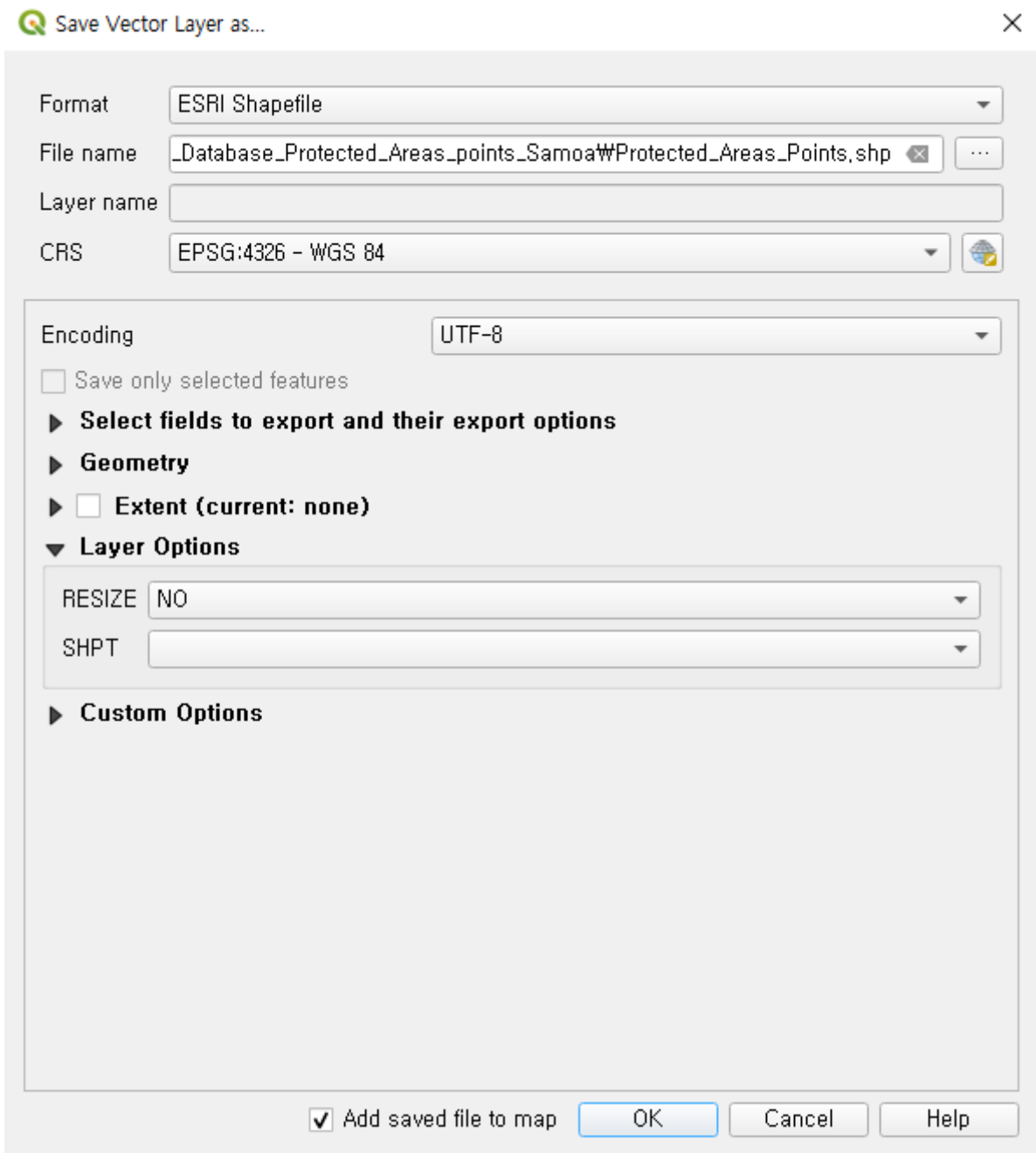


Figure 18 To save vector layer as point shape file

STEP 3. Convert Vector to Raster

1. Click **Processing > Toolbox > SAGA > Raster creation tools > Rasterize**
 - 1) Put **Grid_code** as an Attribute, **attribute** as an Output Value, **Use layer extent** (same as point layer) as a Set output extent, **Distance between points** as a Cell size, **[0]thin** and **[0] node** as Method for Lines and **[0] node** as Fit

2) Save file

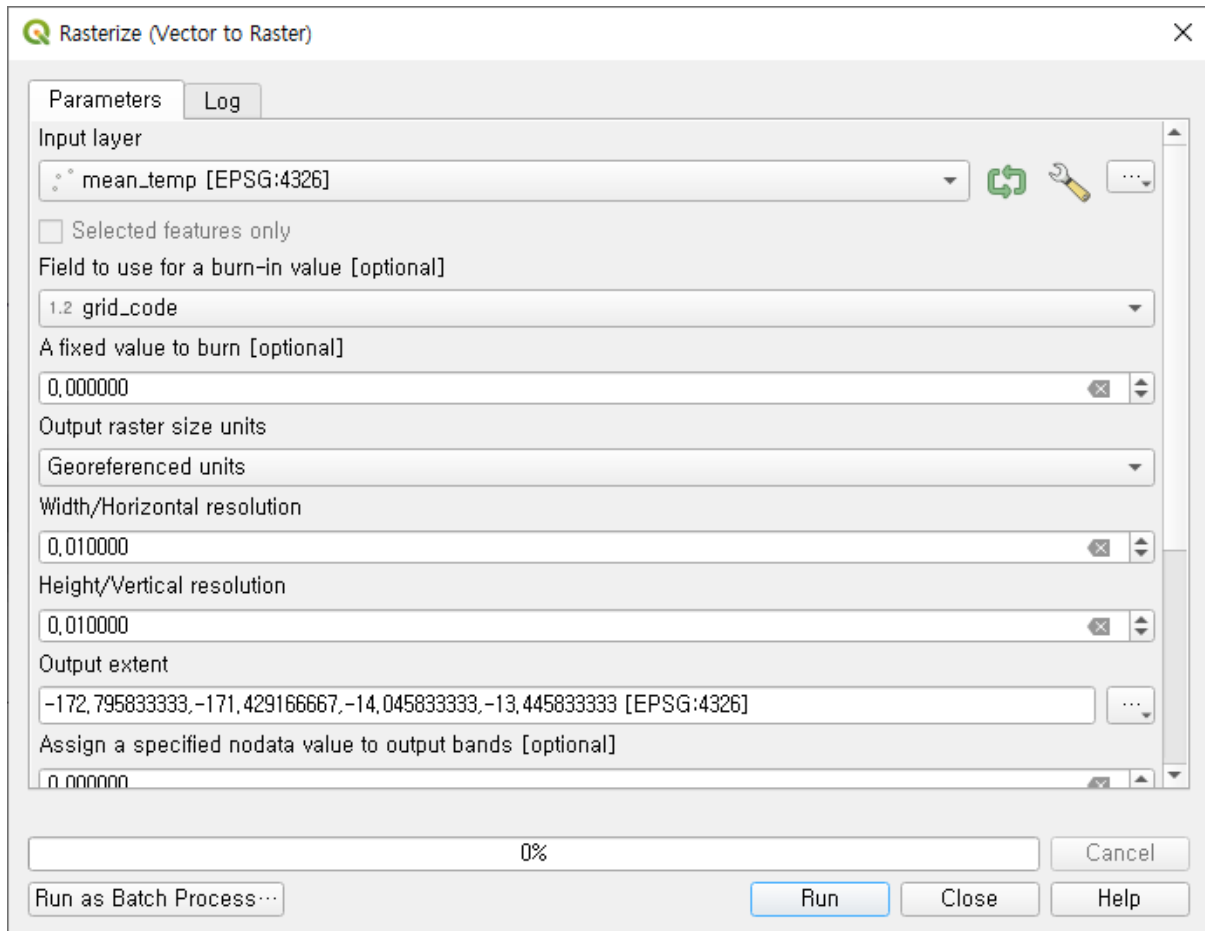


Figure 19 To rasterize



Figure 20 Vector of Samoa

STEP 4. Add Raster Data

1. Click **Layer > Add Layer > Add Raster Layer**
2. Click on the icon (**Data Source Manager**) **Raster > Source > Add data**
3. Open **DEM (... .tif)** raster dataset from source
(./KU_material/1_Background/ DEM/Dem_ky.tif)

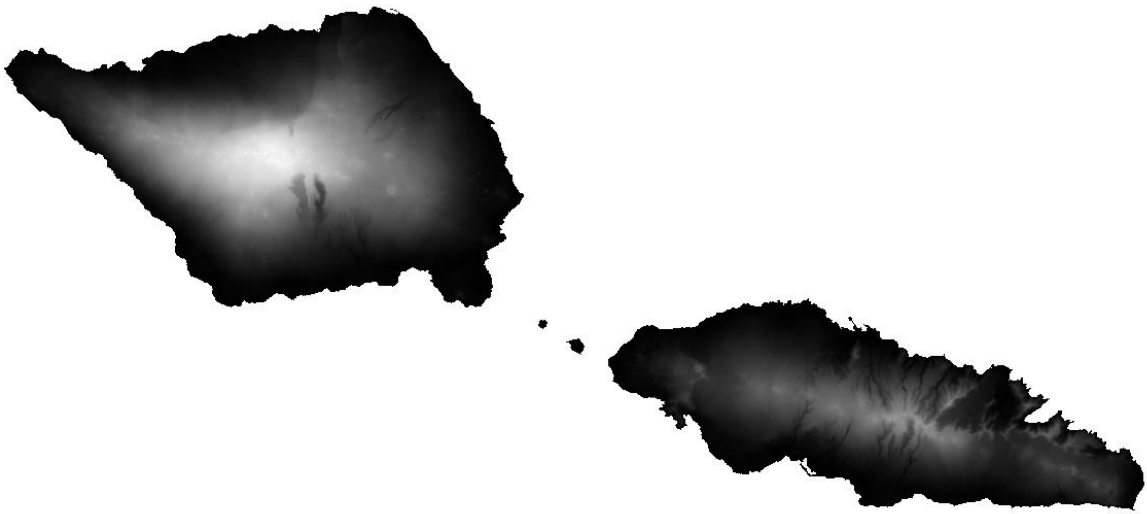


Figure 21 To add raster data

STEP 5. Change Layer Styling

4. Click on the **Property > Symbology**
5. Change Singleband pseudocolor at a Render type, Spectral at color ramp and change min and max value, opacity and custom style etc.

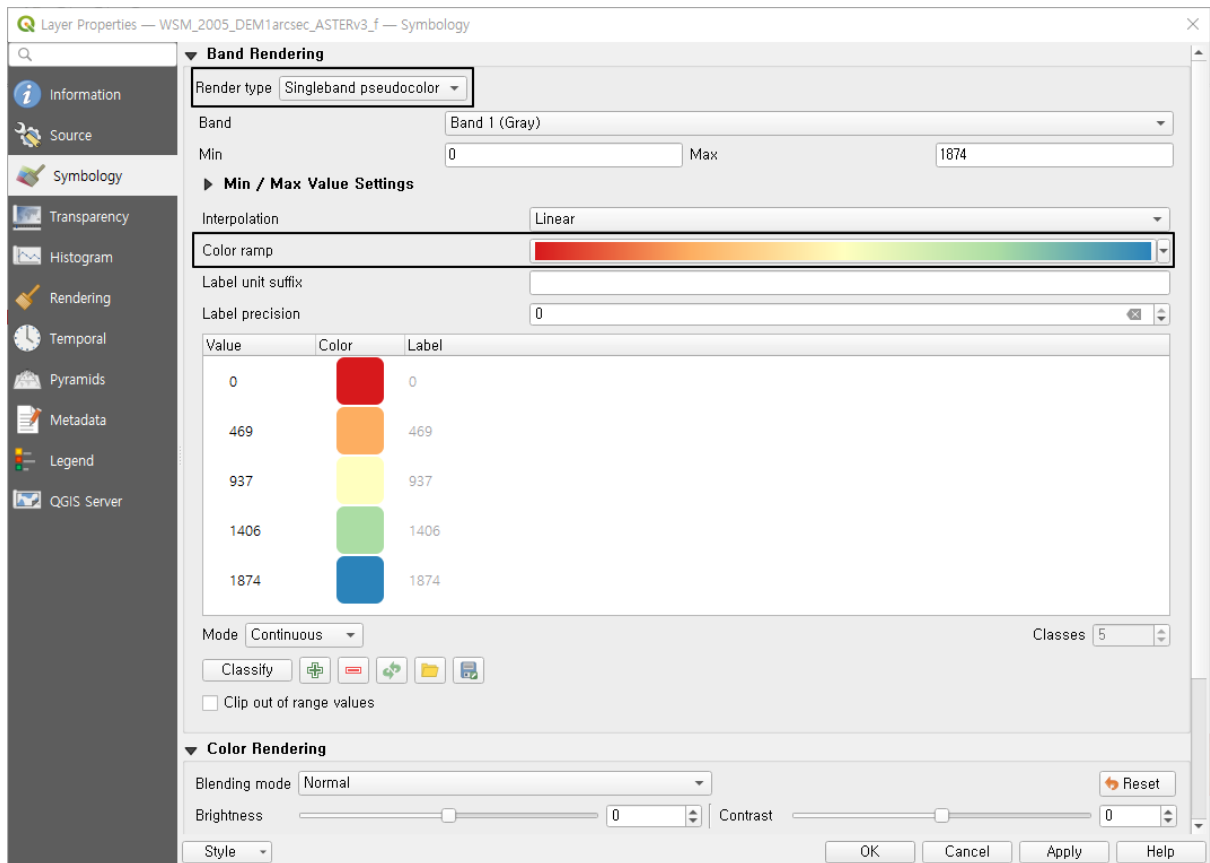


Figure 22 To change layer styling

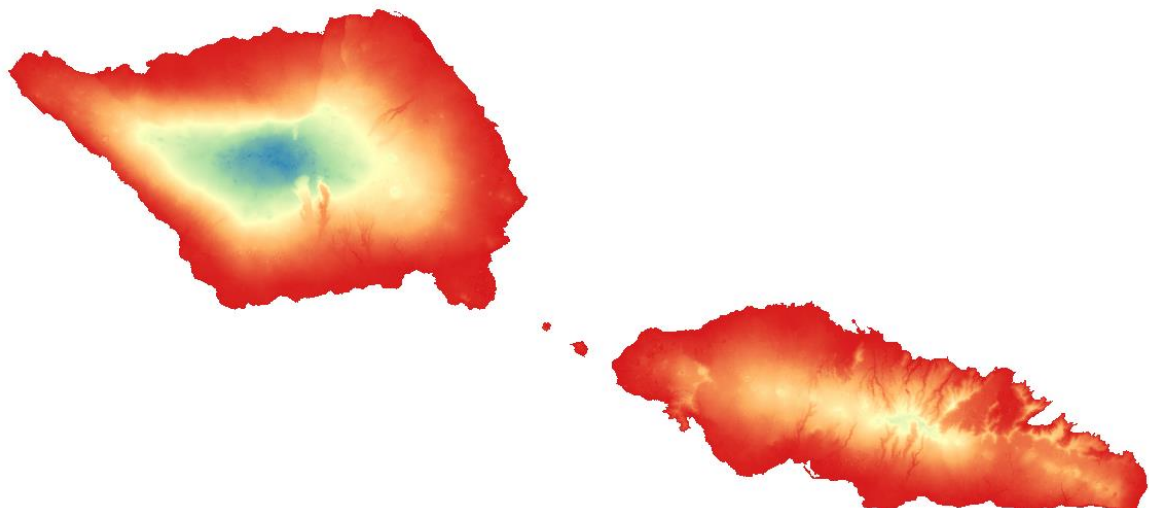


Figure 23 Layer styled raster file

STEP 6. Map Composer

1. Click Project > New Print Layout
2. Click Add item > Add Map, Add Legend, Add Scale Bar, Add North Arrow

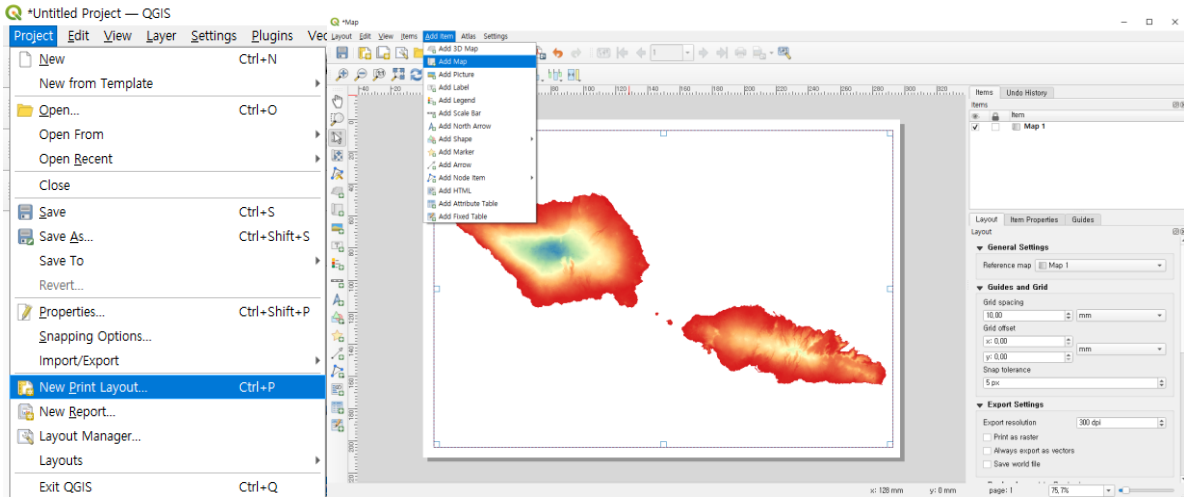


Figure 24 To add map, legend, scale bar, and north arrow

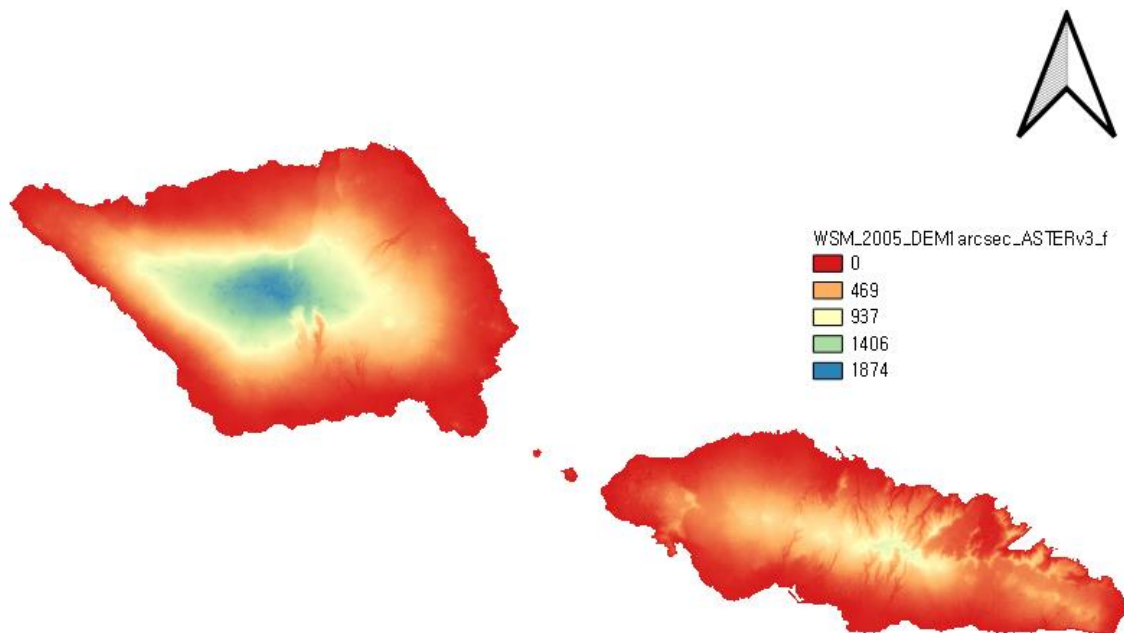


Figure 25 Added map, legend, scale bar, and north arrow

II. Data (Satellite Imagery) Acquisition in GEE

II -1. Introduction of GEE (Google Earth Engine)

Google Earth Engine (GEE) is a cloud-based geospatial analysis platform where users can visualize and analyze massive satellite imagery assets in planetary scale. This platform provides forty years of historical imagery and scientific dataset based on its own public data archive and functions to analyze the data.



Figure 00 Google Earth Engine Logo

Within the archive of GEE, various kind of Earth Observation (EO) data are included: (1) Satellite imagery from different sensors such as Landsat, Sentinel and MODIS (2) Climate and weather data (e.g. surface temperature, sea surface temperature, precipitation data, LAI(Leaf Area Index), etc.) (3) Geophysical Data (e.g. Digital Elevation Models(DEM), land cover maps, cropland data, night-time imagery, etc.)

To use GEE, it is necessary to connect to the service through the API (Application Programming Interface). API (Application Program Interface) is a way of communicating with a particular computer program or internet service and provides a point where user and service can interact. GEE has three different APIs which are JavaScript, Python and REST. Today, we're going to access GEE with JavaScript API using the Code Editor.

As a tip, make sure that you have enough storage space in the Google Drive before getting data from GEE since, in most cases, we need multiple satellite images, and they take up plenty amount of it.

II -1. What is Code Editor

The Earth Engine Code Editor (<https://code.earthengine.google.com/>) is a web-based IDE (integrated Development Platform) for the JavaScript API. To get the access for the Code Editor, it is needed to log in with a Google Account. In Code Editor, we can write the script using JavaScript to search, filter, process, analyze and download the satellite images in the GEE catalog. The elements of Code Editor are shown in the Figure 00 provided by GEE.

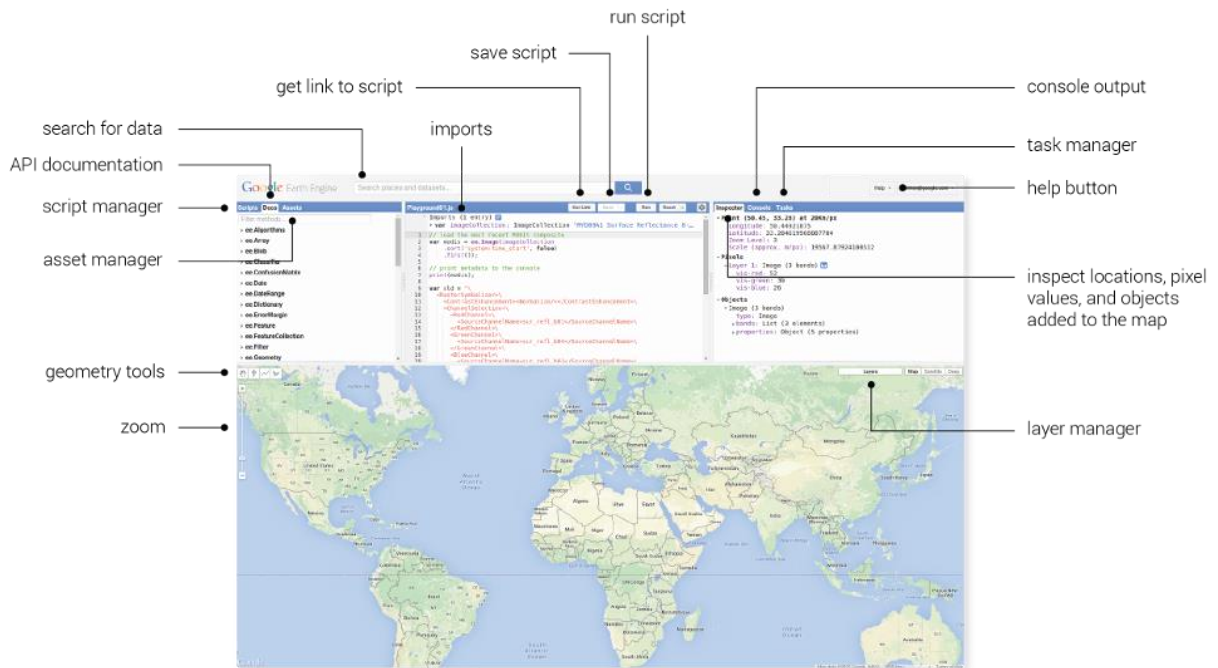


Figure 00 Elements of Code Editor
(Source from Google Earth Engine (GEE))

As our manual is more focused on a land classification task itself than a data processing with GEE, the detailed explanation about Code Editor and JavaScript isn't included. Instead, the simple written codes for each process will be given. If you want to practice Code Editor and JavaScript from the scratch, access to the link below for the official tutorials.

(JavaScript Tutorials of GEE:

<https://developers.google.com/earth-engine/tutorials/community/anomalies-analysis-smo-and-pre>)

II -2. Data Processing with Code Editor

To acquire clear and task-oriented satellite images, it is inevitable to apply techniques to process them. Therefore, Code Editor has variety of functions to do this. First, to sort out the images from the collection, (1) filter images with the required condition and (2) map cloud and cirrus mask, and scale factors. Then, after selecting the bands that are needed, (3) apply multi-temporal composite with median filter within the target time range. Finally, to fill the missing data where clouds and cirrus exist previously, (4) mosaic the composite image in longer period.

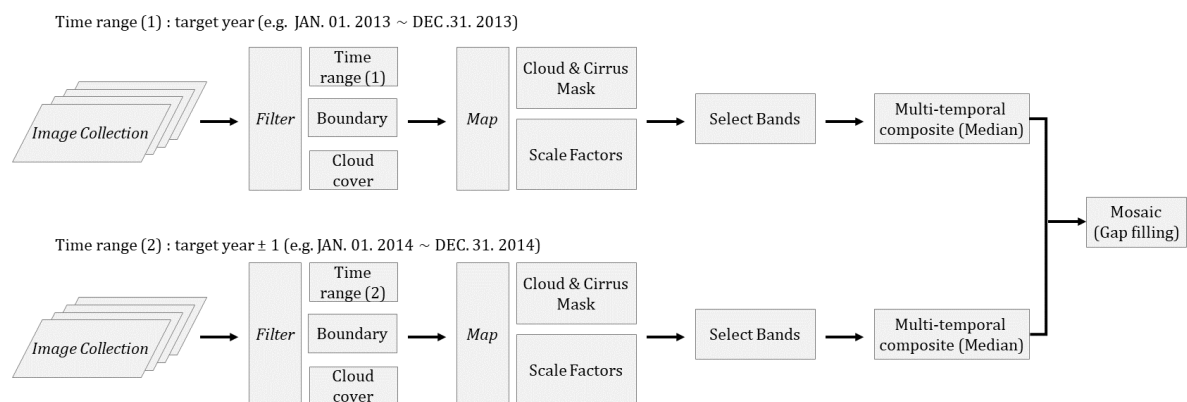


Figure 00 Flowchart of Data Processing with GEE (Google Earth Engine)

The detailed process of each step will be explained in the following section <II -2.1. How to process Landsat Imagery using Code Editor>.

II -2.1. How to Process Landsat Imagery using Code Editor

STEP 1. Load Landsat Imagery with Filters

1. Access to Code Editor (<https://code.earthengine.google.com/>) and log in with your Google Account
2. Upload the boundary of Samoa: Click on **Assets > New > Shape files > Source files > Select**
* Files with extension “.shp, .shx, .dbf” should be uploaded at once
3. Import the **boundary shape file** to the script: Click on **Import into script** of the boundary shape file in the **Assets**
4. Load **Landsat-8 Collection** using the method (`ee.ImageCollection`)

5. Filter the collection by time range, boundary and ratio of cloud cover with chaining the filtering methods (*ee.ImageCollection.filterDate*, *ee.ImageCollection.filter*)
6. Below is the source code for STEP 1

```

Imports (1 entry)
var samoa: Table projects/samoa23/assets/samoa
1 var l8_13_d = ee.ImageCollection("LANDSAT/LC08/C02/T1_L2")
2   .filterDate('2013-01-01', '2013-12-31')
3   .filter(ee.Filter.calendarRange(5, 10, 'month'))
4   .filterBounds(samoa)
5   .filterMetadata('CLOUD_COVER_LAND', 'less_than', 60)

```

Figure 00 Source code to load Landsat imagery with Filters

STEP 2. Map cloud & cirrus mask and the scale factors

1. Define the functions for the cloud & cirrus mask and the scale factors
 * The scale factors and offsets for each band of Landsat 8, Level 2, Collection 2, Tier 1 can be found in Earth Engine Data Catalog (https://developers.google.com/earth-engine/datasets/catalog/LANDSAT_LC08_C02_T1_L2)

Name	Units	Min	Max	Scale	Offset	Wavelength	Description
SR_B1		1	65495	2.75e-05	-0.2	0.425-0.451 µm	Band 1 (ultra blue, coastal aerosol) surface reflectance
SR_B2		1	65495	2.75e-05	-0.2	0.452-0.512 µm	Band 2 (blue) surface reflectance
SR_B3		1	65495	2.75e-05	-0.2	0.533-0.590 µm	Band 3 (green) surface reflectance
SR_B4		1	65495	2.75e-05	-0.2	0.636-0.673 µm	Band 4 (red) surface reflectance
SR_B5		1	65495	2.75e-05	-0.2	0.851-0.879 µm	Band 5 (near infrared) surface reflectance
SR_B6		1	65495	2.75e-05	-0.2	1.566-1.651 µm	Band 6 (shortwave infrared 1) surface reflectance
SR_B7		1	65495	2.75e-05	-0.2	2.107-2.294 µm	Band 7 (shortwave infrared 2) surface reflectance

Figure 00 Scale Factors and offsets of Landsat-8 in Earth Engine Data Catalog

2. Map the defined functions for the cloud & cirrus mask and the scale factor over the collection (*l8_13_d*) using the method (*ee.ImageCollection.map*)

3. Below is the source code for STEP 2

```
8 //Applying the scale factors
9 ▾ function applyScaleFactors(image) {
10   var opticalBands = image.select('SR_B.').multiply(0.0000275).add(-0.2);
11   //var cdistBand = image.select('ST_CDIST').multiply(0.01);
12   //var thermalBands = image.select('ST_B.*').multiply(0.00341802).add(149.0);
13   return image.addBands(opticalBands, null, true);
14   //      .addBands(thermalBands, null, true);
15   //      .addBands(atranBand, null, true).addBands(cdistBand, null, true);
16 }
17
18 //Applying the cloud & cirrus mask
19 ▾ function maskL8srClouds(image) {
20   // Bits 3 and 5 are cloud shadow and cloud, respectively.
21   var cloudShadowBitMask = (1 << 4);
22   var cloudsBitMask = (1 << 3);
23   var cirrusBitMask = (1 << 2);
24   // Get the pixel QA band.
25   var qa = image.select('QA_PIXEL');
26   // Both flags should be set to zero, indicating clear conditions.
27   //kernel
28   var kernel_radius = 2
29   var kernel = ee.Kernel.square({
30     radius: kernel_radius, units: 'pixels', normalize: false
31   });
32   var kernel_sum = Math.pow(kernel_radius*2+1,2)
33
34   var mask = qa.bitwiseAnd(cloudShadowBitMask).eq(0)
35   //      .and(qa.bitwiseAnd(cloudsBitMask).eq(0))
36   //      .and(qa.bitwiseAnd(cirrusBitMask).eq(0)).convolve(kernel).eq(kernel_sum);
37   return image.select('SR_B.', 'QA_PIXEL')
38 }
39 //map the functions
40 l8_13_d.map(maskL8srClouds)
41 //      .map(applyScaleFactors);
```

Figure 00 Source code to map cloud & cirrus mask and the scale factors over the collection

STEP 3 & 4. Multi-temporal composite with median filter

& Mosaic composite images for gap filling

1. Composite all the images in the collection (*l8_13_d*) of the target year using the median filter method (*ee.ImageCollection.median*) * the target year of the source code is '2013'
2. Execute the same process (STEP1 & 2) on the collection in longer period for gap filling * the year for gap filling in the source code is from 2014 to 2016
3. Composite all the images in the collection (*l8_13_d_backup*) for gap filling using the median filter method (*ee.ImageCollection.median*)
4. Mosaic the composite images of the target year (*composite_l8_13_d*) and for gap filling (*composite_l8_13_d_backup*)

5. Below is the source code for STEP 3 & 4

```
41 var 18_13_d_backup = ee.ImageCollection("LANDSAT/LC08/C02/T1_L2")
42   .filterDate('2013-01-01', '2016-12-31')
43   .filter(ee.Filter.calendarRange(5, 10, 'month'))
44   .filterBounds(samoa)
45   .filterMetadata('CLOUD_COVER_LAND', 'less_than', 60)
46   .sort('CLOUD_COVER_LAND')
47   .map(maskL8srClouds)
48   .map(applyScaleFactors);
49
50 var composite_18_13_d_backup = 18_13_d_backup.median().clip(samoa);
51 var 18_21_d_expand = ee.ImageCollection.fromImages([composite_18_21_d_backup, composite_18_21_d]).mosaic()
```

Figure 00 Source code to composite images with median filter and mosaic composite images for gap filling

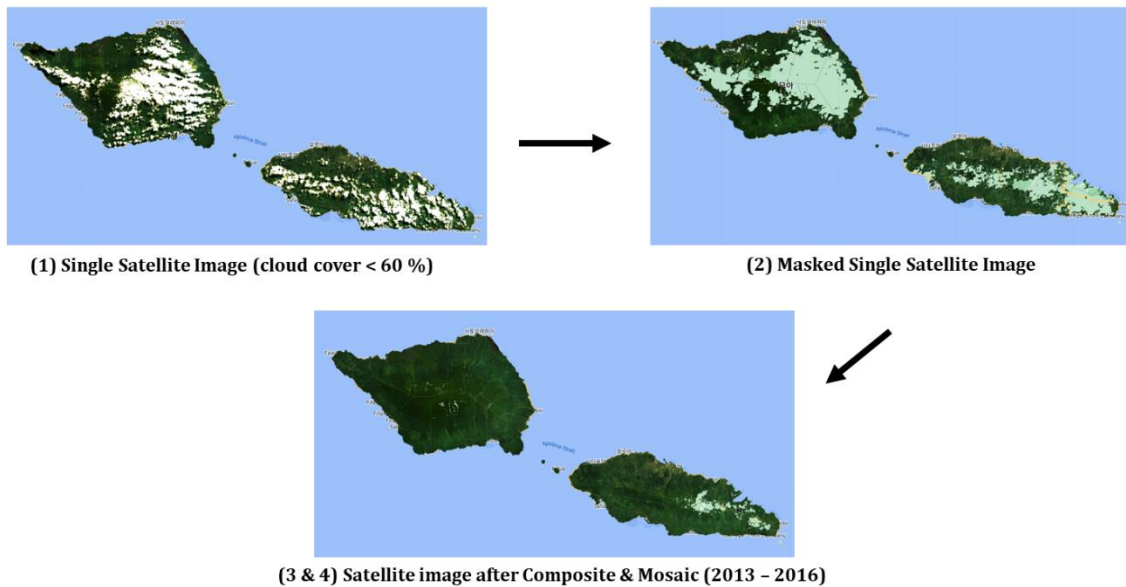


Figure 00 Changes in Satellite Image during the Data Processing

In case of Samoa where high-quality of satellite imagery are insufficient due to the occurrence of persistent noises around the cloud forest, it is recommended to process data with further techniques to remove all the pixels with cloud and cirrus clearly. To know more about the data processing, feel free to view the full version of the source code using link below.

(Link for the source code full ver.:

<https://code.earthengine.google.com/33f3c2351b237113ebdef012dd78264f>)

III. Classification

III -1. Introduction of Classification

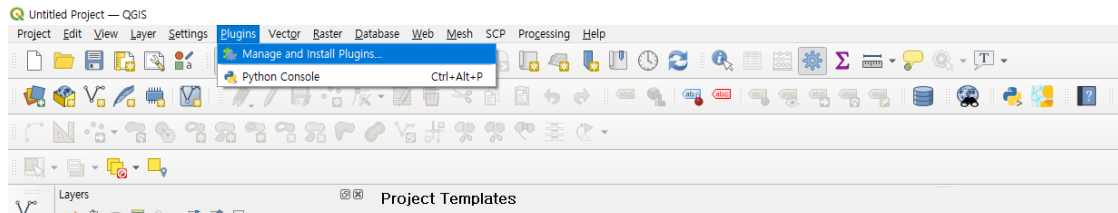
Classification involves the process of categorizing each pixel into predefined classes. Methods for classification are typically categorized into two main types: supervised and unsupervised classification. Unlike unsupervised classification, supervised classification requires input from the user based on predefined criteria.

	Supervised learning	Unsupervised learning
Definition	Training the classifier with labeled training data where the correct answers (classification categories) are specified.	Training the classifier with unlabeled training data where the correct answers are not specified.
Learning Data	Training data with specified labels (correct answers).	Training data without specified answers (labels).
Learning Process	Train the classifier with the training data and then validate it using the validation data.	Using clustering algorithms to group pixels or objects that have similar characteristics together.
Class	Information about the classification categories is needed in advance.	Discovering classification categories through the data.
Application	Suitable for classifying images into specific categories.	Suitable for Exploratory Data Analysis (EDA).
Other characteristics	<ul style="list-style-type: none"> - Low flexibility, high complexity. - Relatively high level of user involvement. 	<ul style="list-style-type: none"> - High flexibility, low complexity. - Relatively low level of user involvement.
Representative algorithm	Maximum Likelihood, Random Forest, SVM (Support Vector Machine), and other ensemble models in machine learning	K-means Clustering, ISODATA

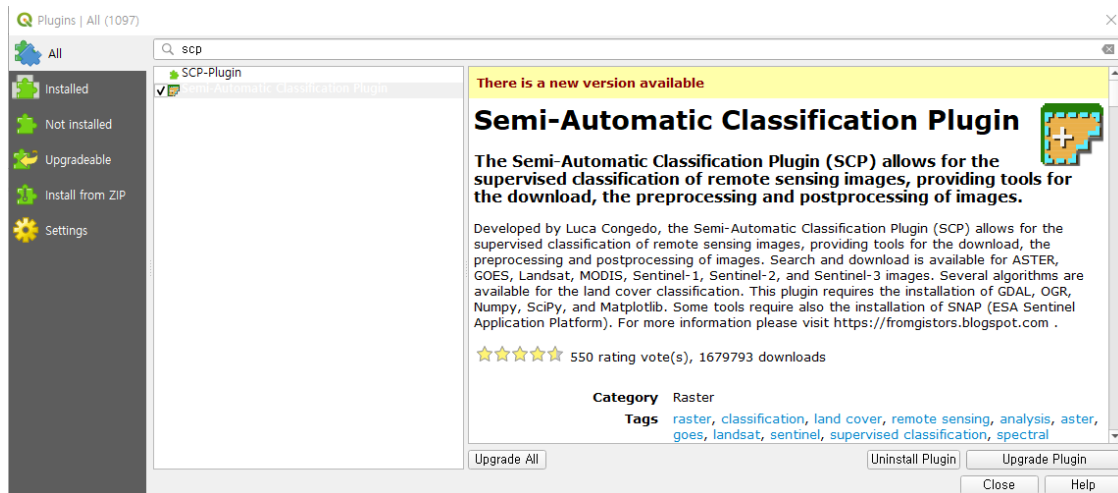
Although QGIS does not offer an integrated classification tool, users can employ the Semi-Automated Classification Plugin (SCP Tool), developed externally, to perform classification tasks.

* Install SCP Tool

1. Click Plugins > Manage and Install Plugins...



2. Search SCP > Add Click Install Plugin



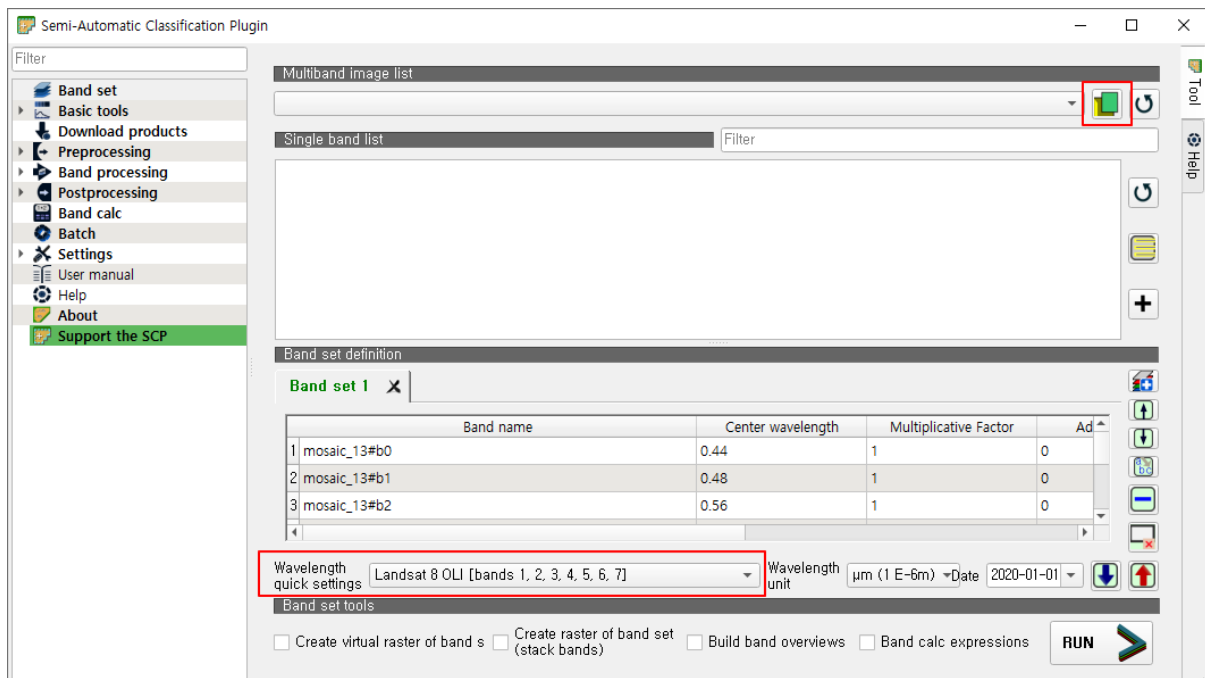
III -2. Class, ROI (Region of Interest)

To classify satellite imagery, one must first determine which classes (or categories) to use. Unlike unsupervised classification, supervised classification requires determining classification categories and setting Regions of Interest (ROIs) based on these categories. Here, an ROI refers to a specific area within the imagery that the user has outlined and defined as one of the predefined categories.

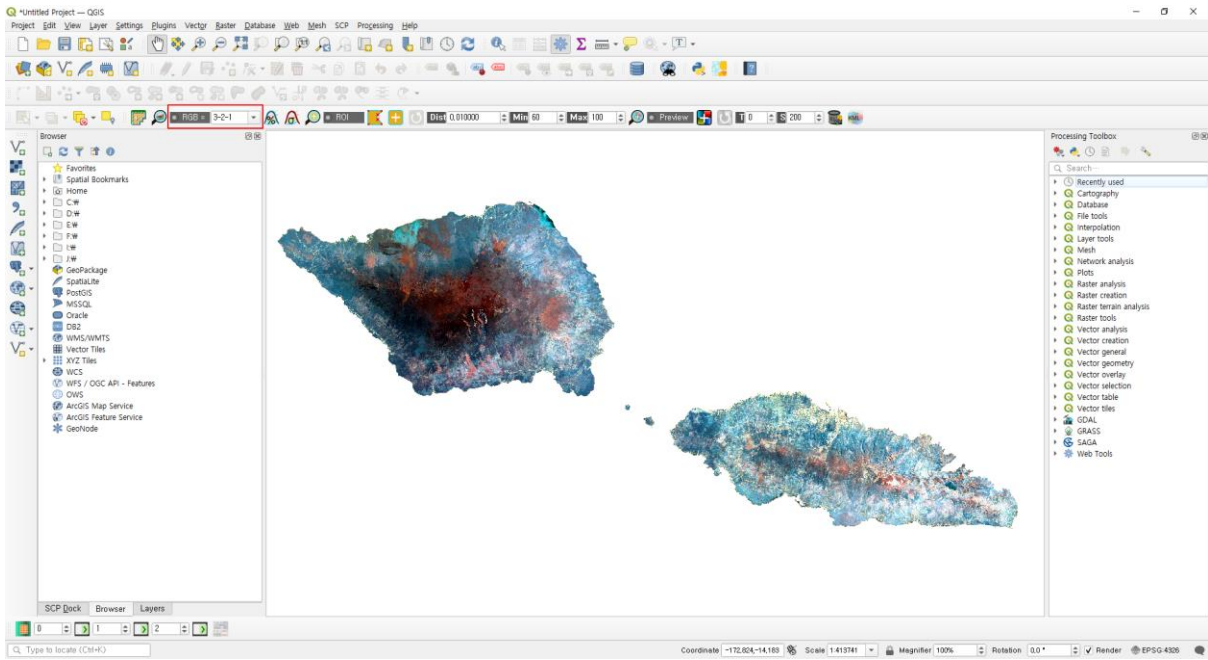
This task involves collecting Training Samples, also known as training data, to generate data that the program can use for classification. For instance, if one wants to classify forests within satellite imagery, the user would manually designate representative areas within the forest and non-forest regions. The program would then be trained using these designated forest and non-forest ROIs, enabling it to automatically classify similar areas as either forested or non-forested based on their data values.

III -3. Supervised Classification Practical Exercise:

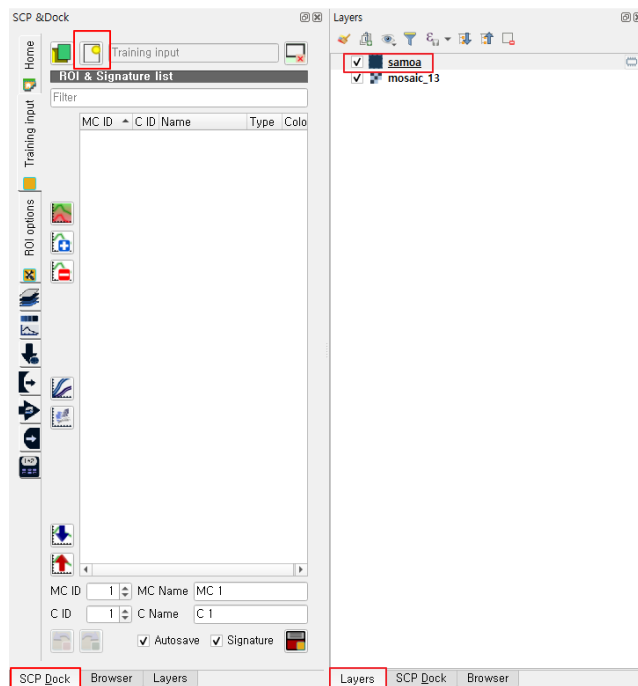
1. Click on the 'Semi-Automatic Classification Plugin' icon, then click on the icon (Open a file) in the Band Set tab. Select the band set you previously created and choose the corresponding Wavelength quick settings. In Wavelength quick setting, you can choose the setting selection according to the satellite imagery you are using.



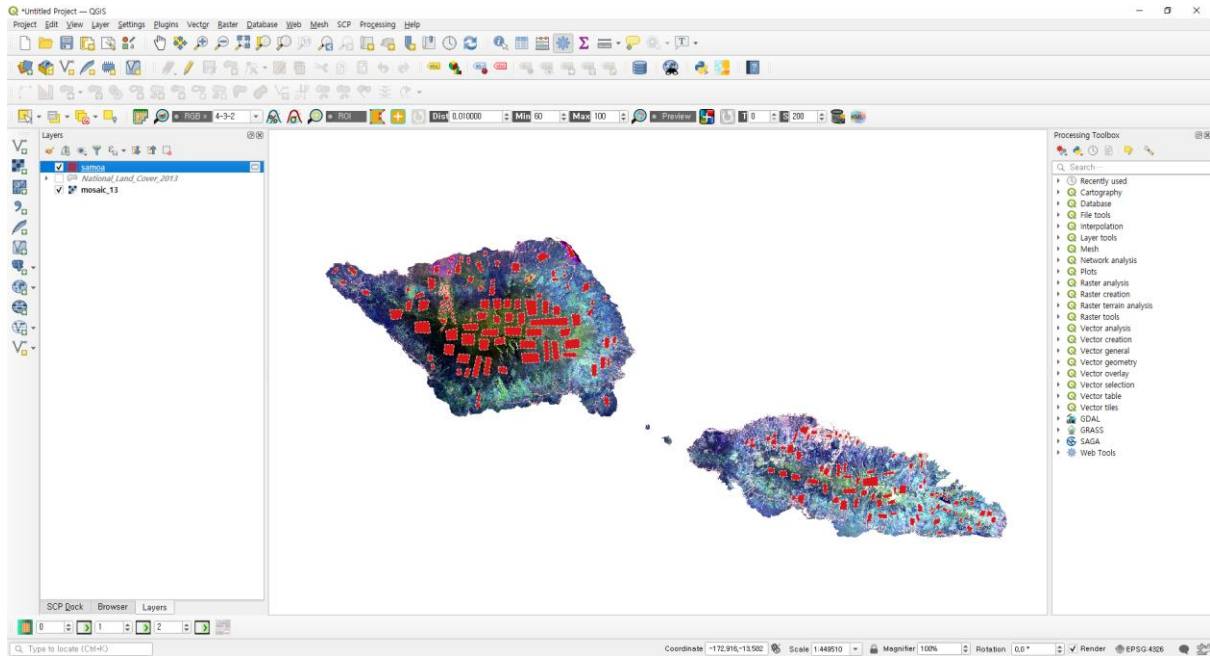
2. On the 'SCP Working Toolbar,' set the 'RGB =' to 3-2-1.



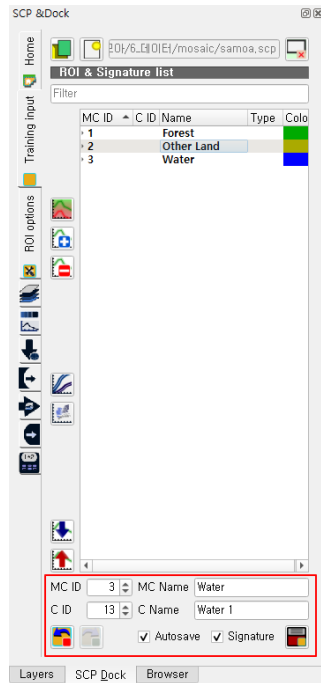
3. After creating the samoa.scp file through the 'Training input' tab in the SCP Dock, verify that the 'samoa' layer has been added to the Layers panel.



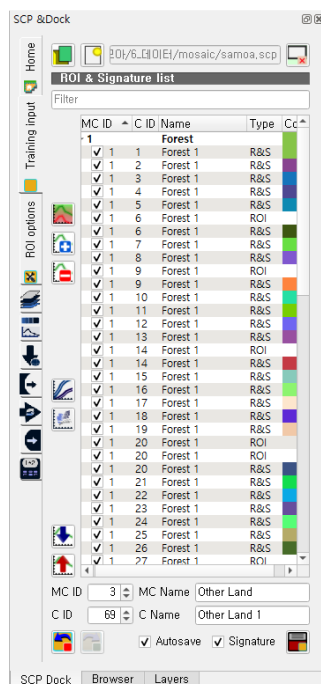
4. In the SCP Working Toolbar, click on 'Create a ROI polygon' and designate the areas corresponding to forests as follows. By changing the previously set RGB values to 4-3-2, areas where vegetation is distributed appear in red, making it easier to distinguish forested areas. When drawing the ROI, left-click to create points and right-click to create polygons.



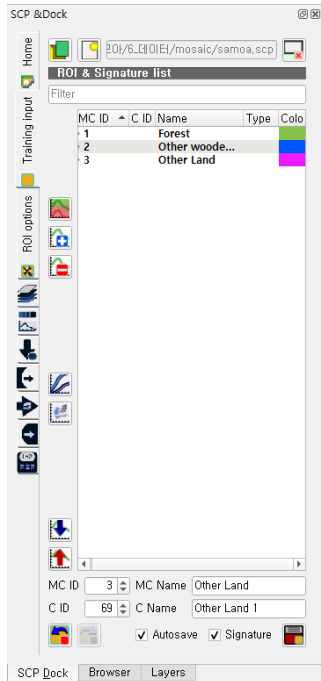
- At the bottom of the SCP Dock panel, there is a section for defining the MC (Macro Class) and C (Sub-Class) factors related to the currently selected area. Here, MC refers to the macro class, while C indicates the sub-class. Information for MC and C is written as illustrated below. Click on 'Save temporary ROI to training input' to create the classes.



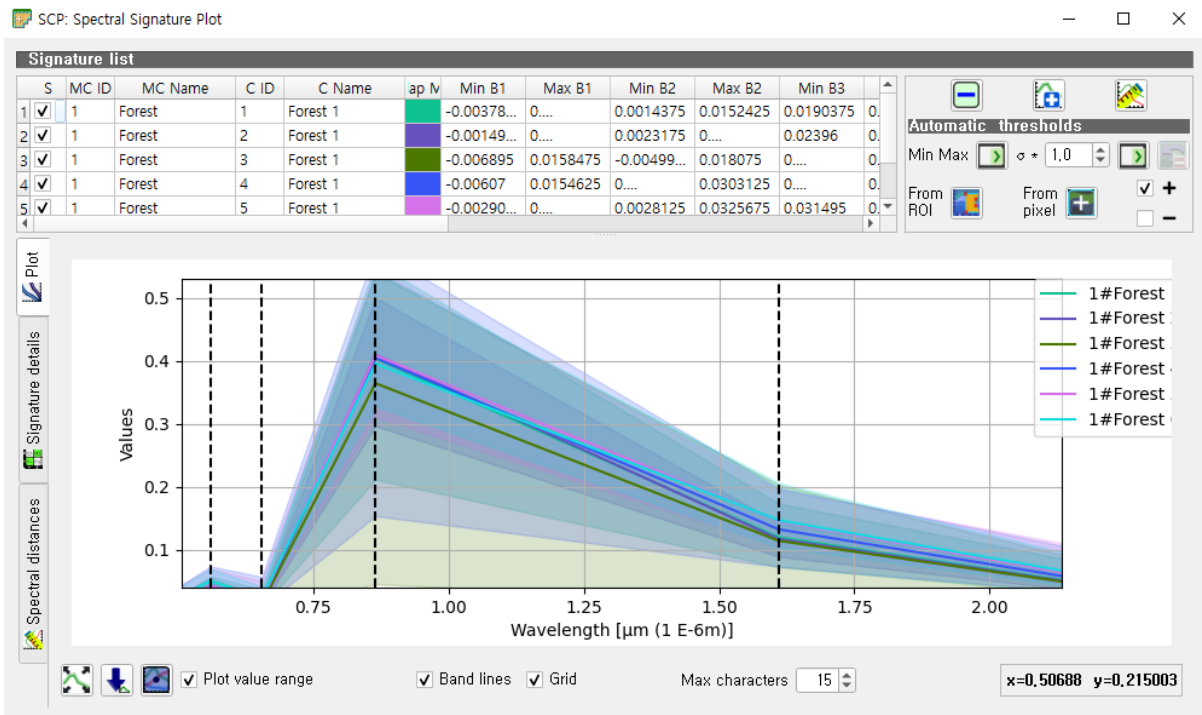
- Set up multiple Classes within a single Macroclass to prepare classification samples. Similarly, add training data for urban areas (Settlement) and water bodies (Water) using the same approach.



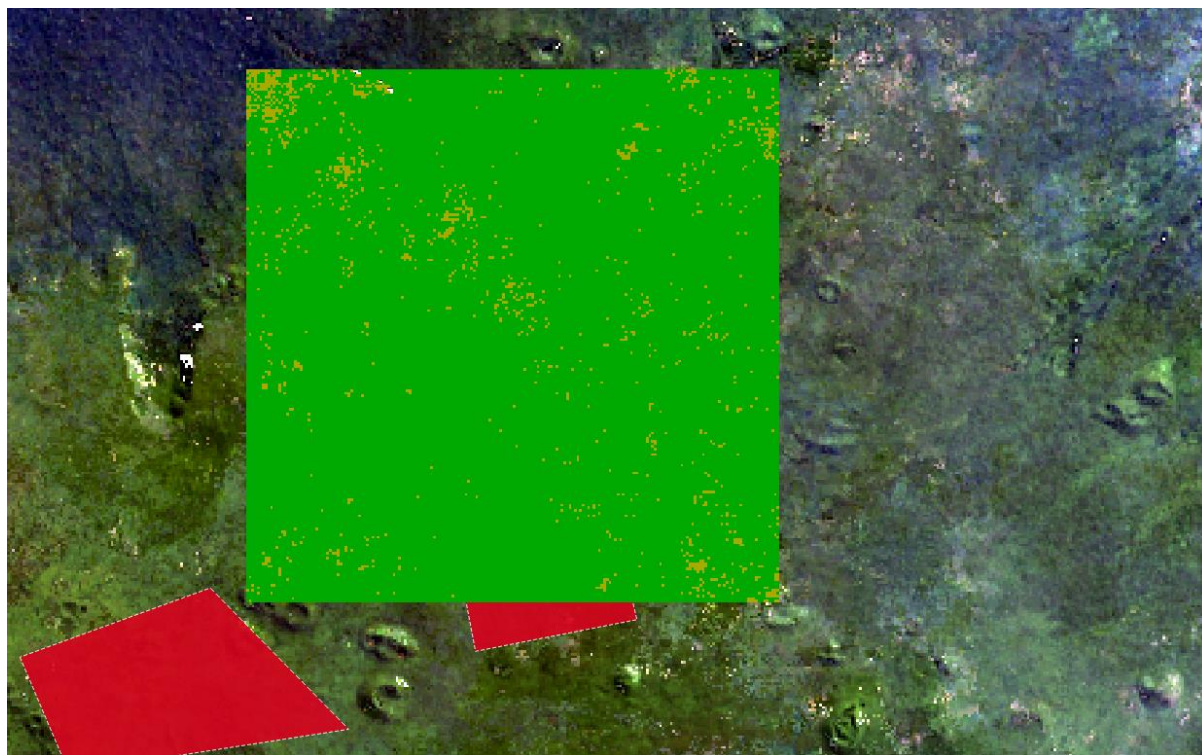
7. In the 'Macroclass list' tab, select appropriate colors for each class.



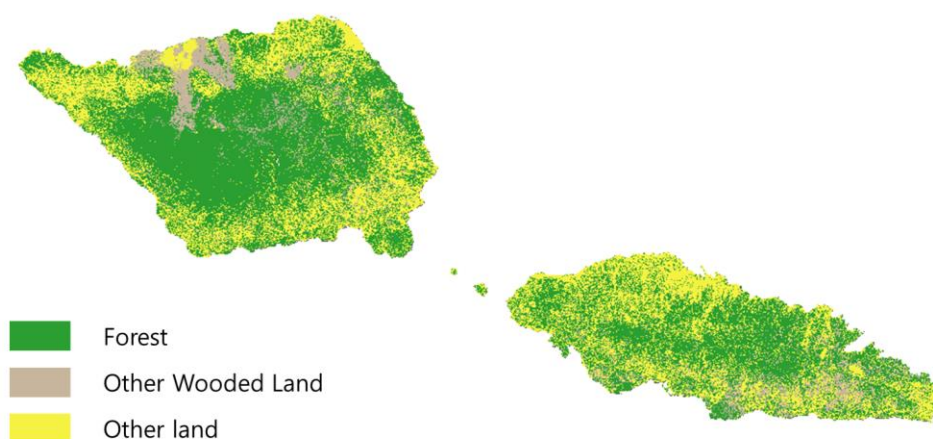
8. In the 'Spectral Signature Plot()' on the left side of the SCP ROI list, you can view the band-wise spectral value statistics and graphs for the ROI areas.



- Click on the 'Classification' tab. Under 'Use', select 'MC ID', and for the 'Algorithm', choose 'Maximum Likelihood'.
- Click on the 'Active classification preview pointer' icon, and when you click on a location in the map view, a classification preview for a certain area will be executed.



- After confirming that the classification is appropriately performed through the preview, execute the 'RUN' command to generate the final result with the classified land cover.

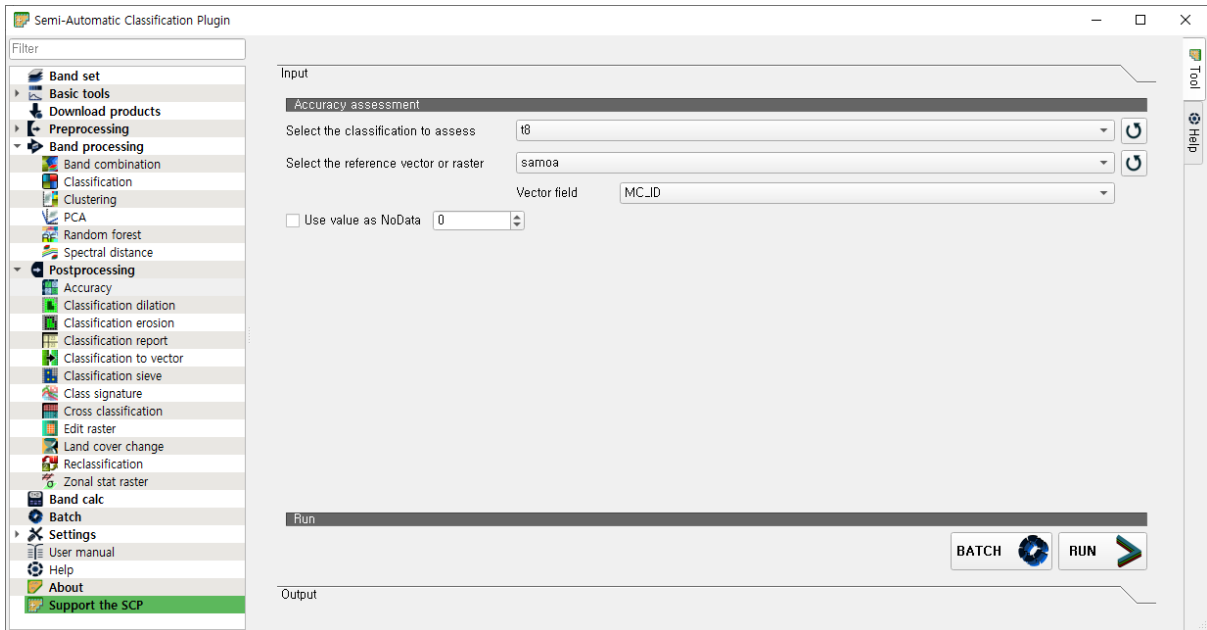


- In the [SCP pop-up window] - [Postprocessing] - [Accuracy] - [input] - [Accuracy assessment], input the following:

Select the classification to assess': t8 (Final result)

Select the reference vector to vector or raster': samoa.scp(*.scp file where the ROI is stored)

'Vector field': MC_ID



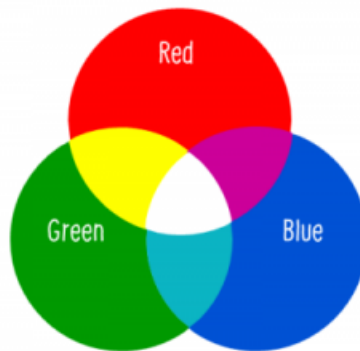
III -4. Unsupervised Classification Practical Exercise:

1. Click on the 'Semi-Automatic Classification Plugin' icon, then click on the icon in the Band Set tab. Select the B2~B7.tif file and choose the corresponding Quick wavelength settings (Landsat 8 OLI [bands 2, 3, 4, 5, 6, 7]).
2. In the left-hand list pane, select 'Band processing', and then choose 'Clustering' at the top.
3. Set the configuration values in the middle Input window as shown in the picture, and then press the RUN button located at the bottom to save the settings.
4. Check the classified output results below to identify which categories the classified numbers correspond to.
5. In 'Postprocessing', select the file you wish to reclassify under 'Reclassification'.
6. Click on the + icon on the right side of the 'Values' to add items. Enter the existing classification values and the new values you want to assign, then press RUN.
7. Change the colors of each classification value in the attribute's symbol to derive the final result.
8. Compare the final output with the previously generated .scp file to calculate accuracy and interpret the results.

III-1. Introduction of Satellite Image Process

III-1.1. Spectral Characteristics

All of the colors on a computer screen, television, or any other display device are comprised of a combination of red, green, and blue light. These are called the primary colors (Red, Green and Blue; RGB) (Figure 39) because all colors that we can see are made using these wavelengths of light.



**Figure 26 The three primary colors of RGB Color Model (Red, Green and Blue)
(Source from Presentitude, 2020, Jan. 20)**

The reflectance values of different surfaces can be plotted as functions of wavelength called "spectral response curves" or "spectral signatures". Differences among the spectral signatures of landscape features are used to classify remotely sensed regions, since the spectral signatures of similar features have similar shapes (Figure 40).

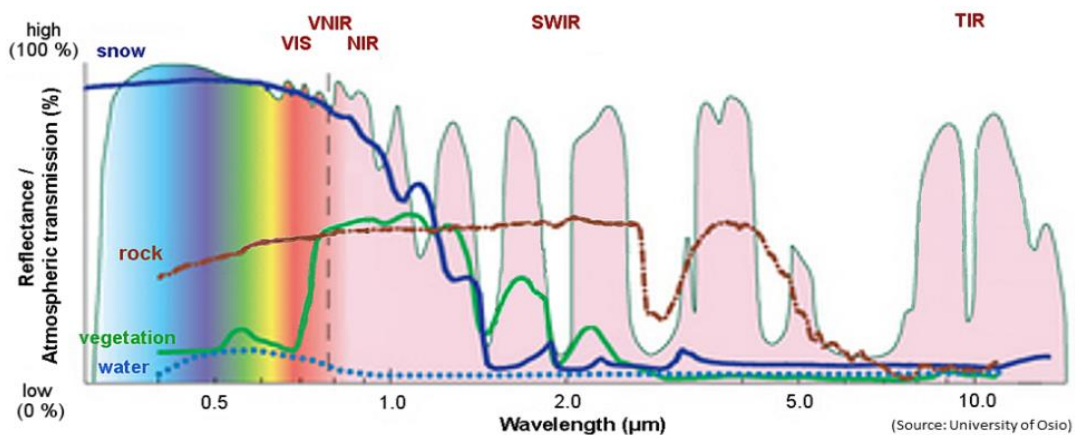


Figure 27 Reflectances of different surfaces (Source from EUMeTrain, 2014)

III-1.2. Band Composition

Selecting the appropriate bands to use in the color image on the other hand does have a huge impact on which features can be seen in a particular image (Figure 41, 42). The list below explains some of the features of the seven Landsat Thematic Mapper bands and how they are tailored for detecting different features.

- Band 1 (0.45-0.52 μm , blue): monitor sediment in water, mapping coral reefs, and water depth
- Band 2 (0.52-0.60 μm , green): monitor green (vegetation)
- Band 3 (0.63-0.69 μm , red): distinguishing between vegetation and soil and in monitoring vegetation health
- Band 4 (0.76-0.90 μm , near infrared): defining the water/land interface
- Up to band 7

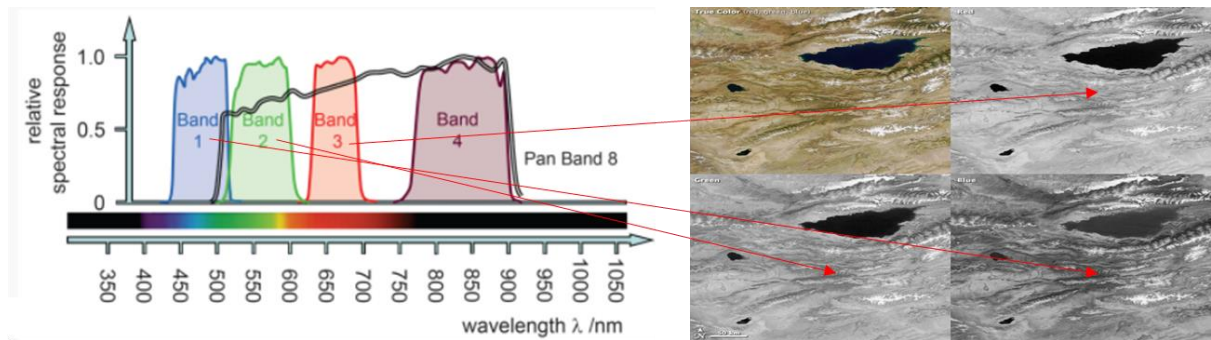


Figure 28 Band composition (spectrum) of satellite image
 (Source from Carl von Ossietzky University of Oldenburg and Matt Rackcliff Landsat 5 data, NASA)

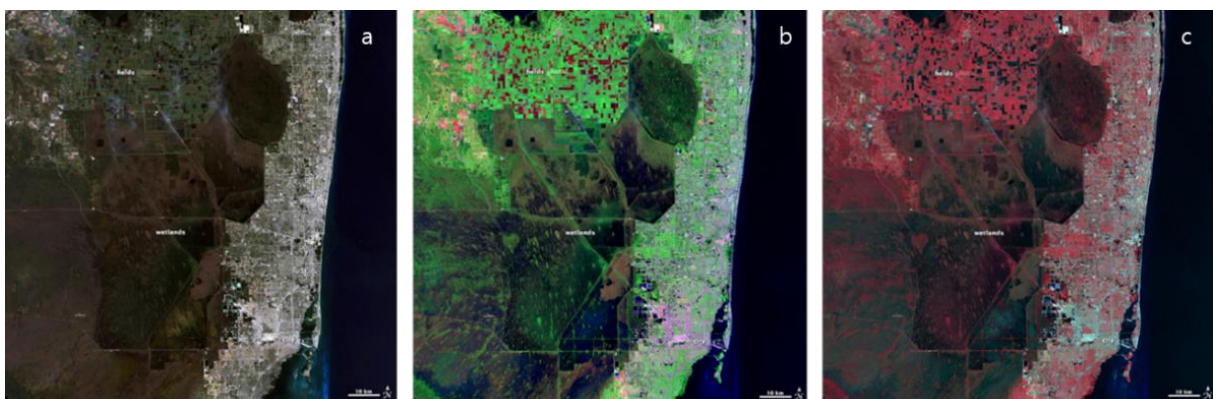


Figure 29 Different color mix to detect different features
 (a. True color - R: Red, G: Green, B:Blue, b. False color for detecting bare soil - R: SWIR, G: NIR, B: Green, and c. False color for detecting vegetation - R: NIR, G: Red, B: Green

(Source from Earth imaging journal, 2020, Jan. 20)

III-2. Satellite Data Processing for Drought Monitoring

III-2.1. Normalized Difference Vegetation Index (NDVI)

NDVI quantifies vegetation by measuring the difference between near-infrared (which vegetation strongly reflects) and red light (which vegetation absorbs) (Figure 43). NDVI always ranges from -1 to +1. But there isn't a distinct boundary for each type of land cover. For example, when you have negative values, it's highly likely that it's water. On the other hand, if you have a NDVI value close to +1, there's a high possibility that it's dense green leaves. But when NDVI is close to zero, there isn't green leaves and it could even be an urbanized area (Figure 44).

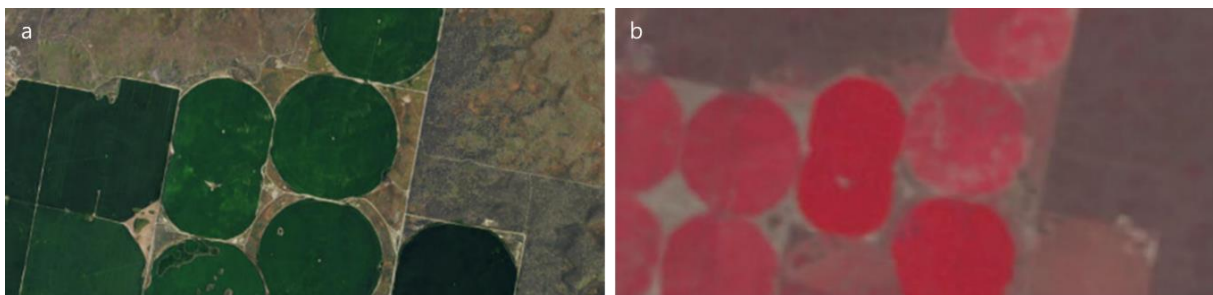


Figure 30 Near-infrared and red light; a. True color - R:Red, G:Green, B:Blue, and b. Normalized difference vegetation index (Source from GIS Geography)

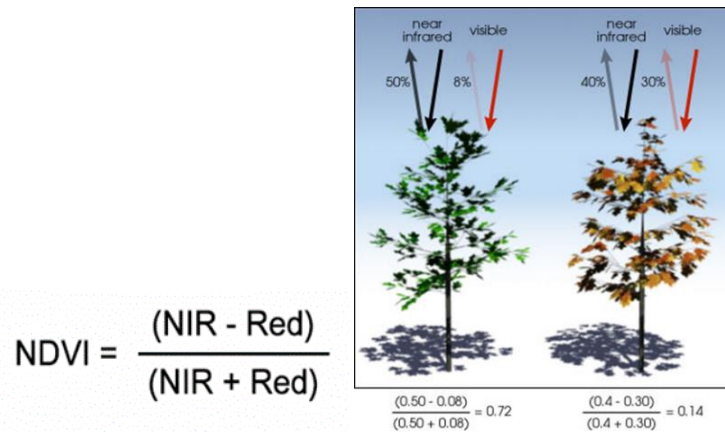


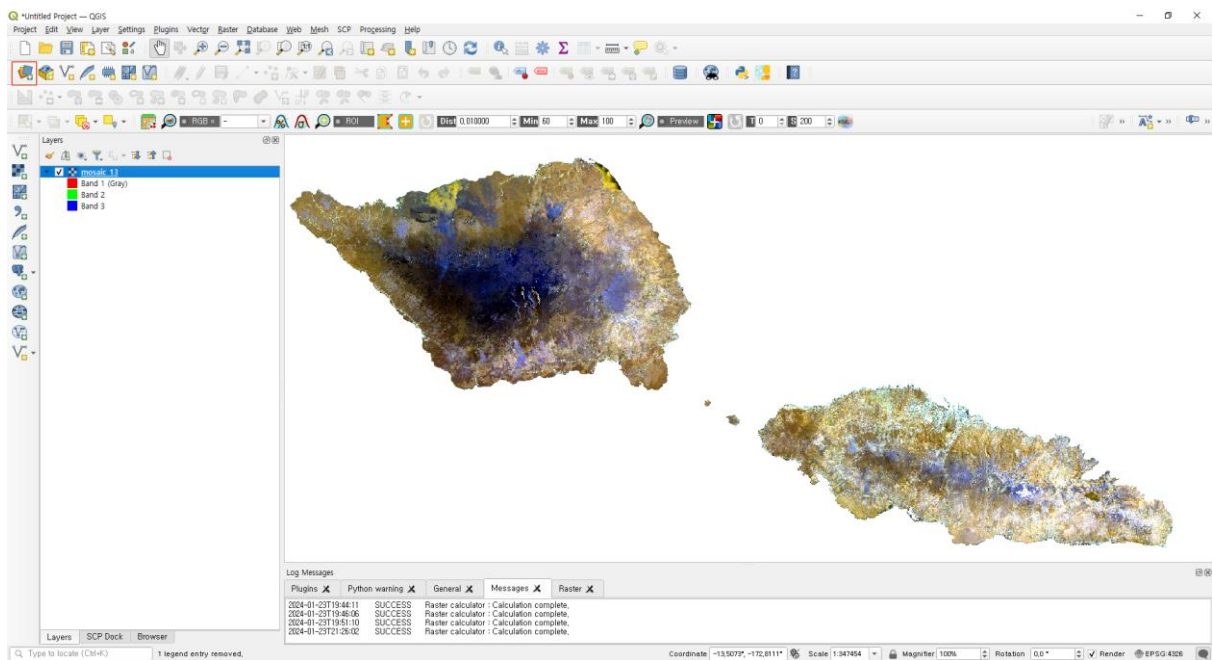
Figure 31 Calculating healthy vegetation (left), and unhealthy vegetation (right) (Source from GIS Geography, 2020, Jan. 20)

III-2.1.1. How derive NDVI with QGIS

STEP 1. Making true and false color images from satellite image (Practice example)

1. Click **Layer > Add Layer > Add Raster Layer**

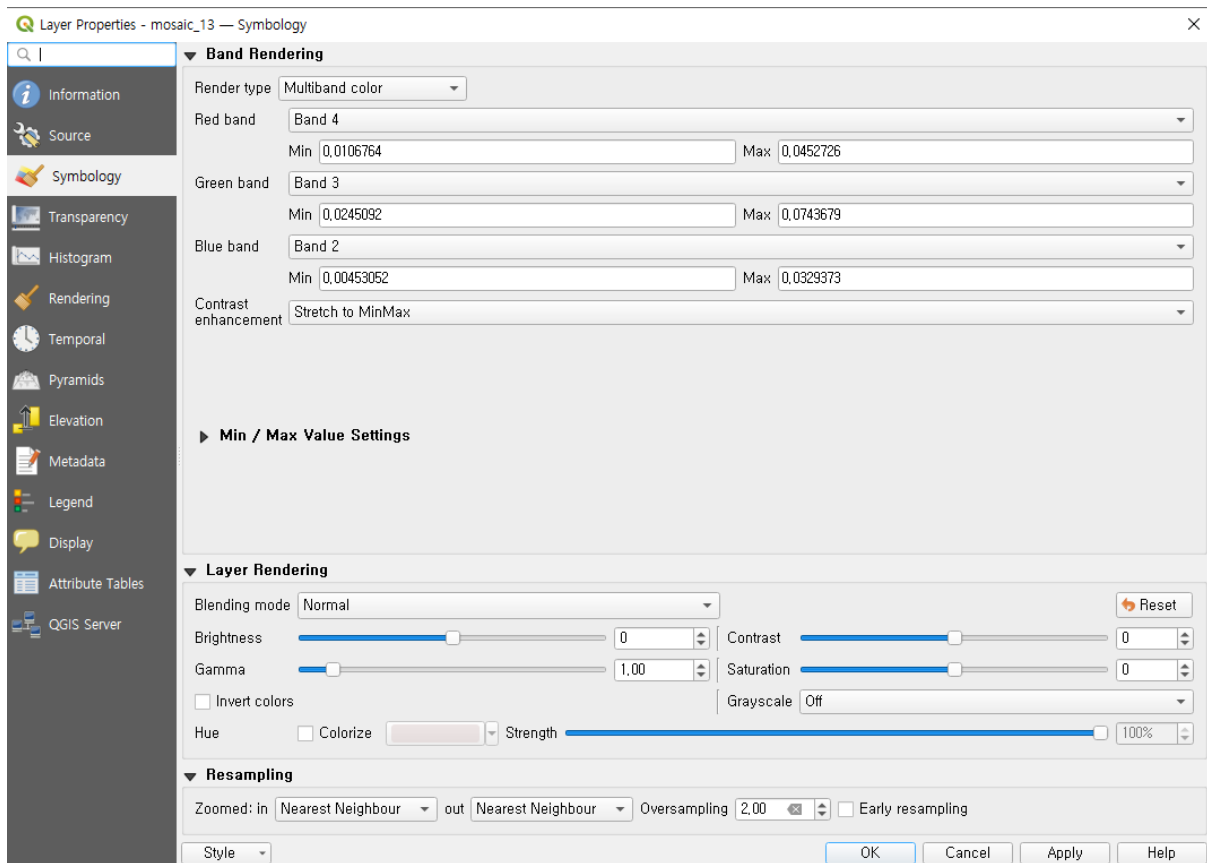
1) Open the file name as **(./mosaic 13 (... .tif))**



2. Set multiband color to make a true color image (LANDSAT-8)

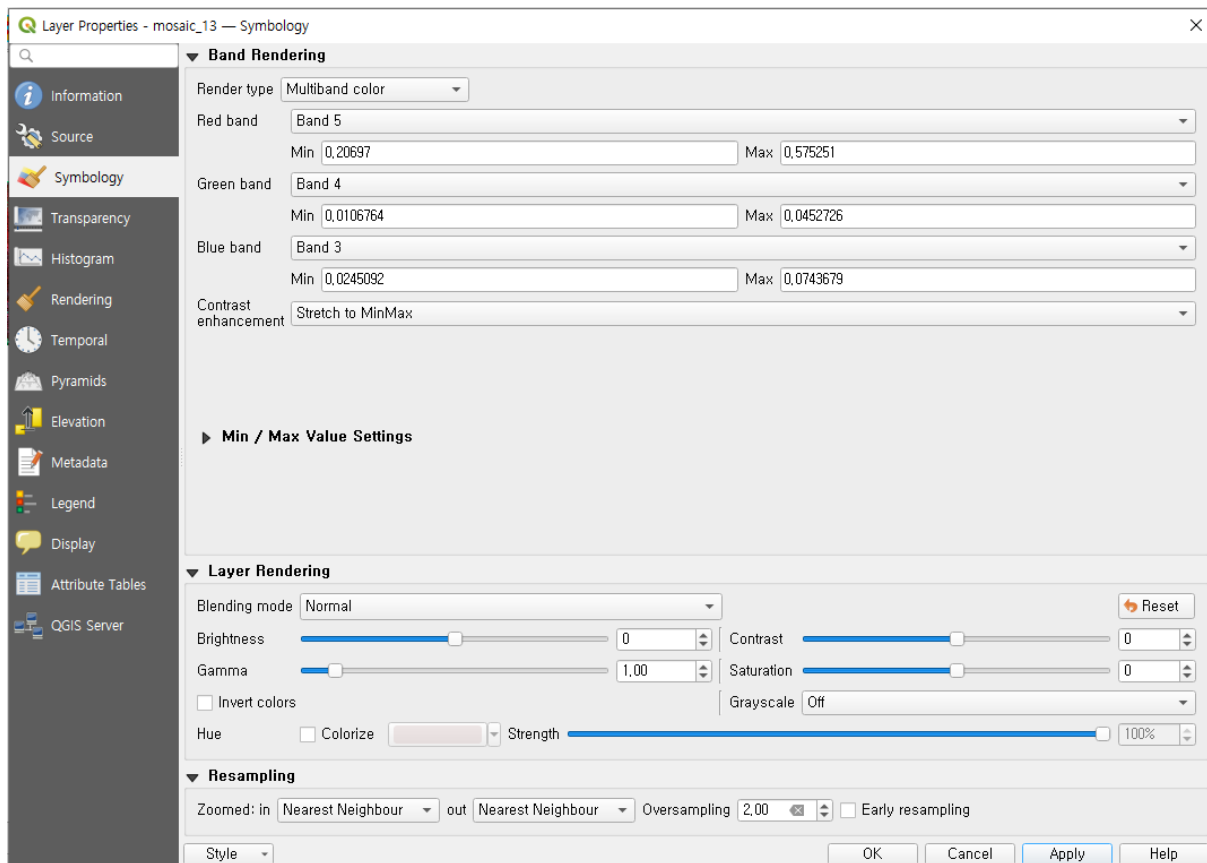
1) Open **Properties > Symbology**

2) Select **Band 4** as a Red band, **Band 3** as a Green band, and **Band 2** as a Blue band



3. Set multiband color to make a false color image

- 1) Select **Band 5** as a Red band, **Band 4** as a Green band, and **Band 3** as a Blue band



STEP 2. Producing NDVI using band characteristics (Practice example)

1. Click **Processing > Toolbox > Raster analysis > Raster calculator**

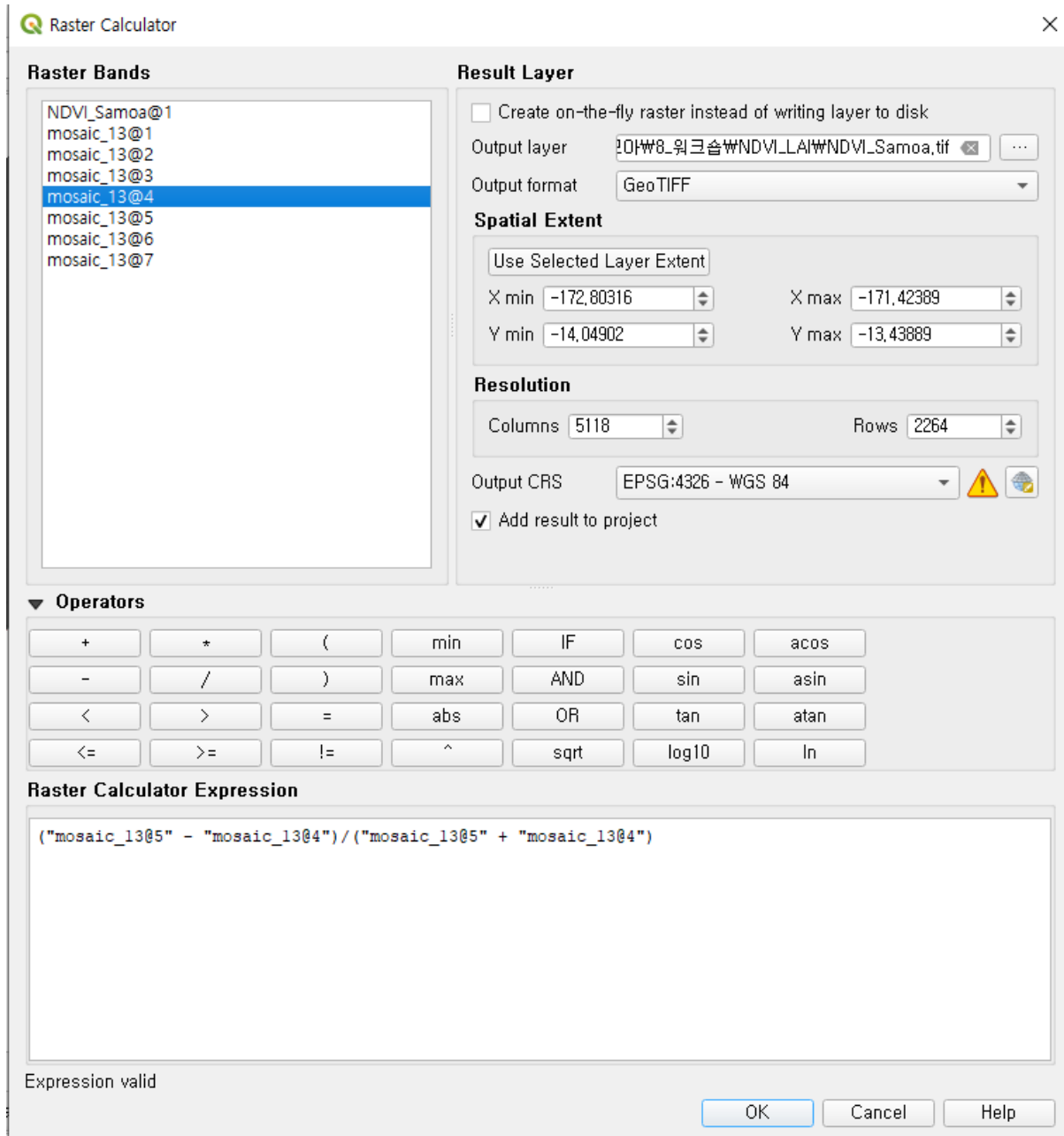
1) Put

$(\text{"mosaic_13@5"} - \text{"mosaic_13@4"}) /$

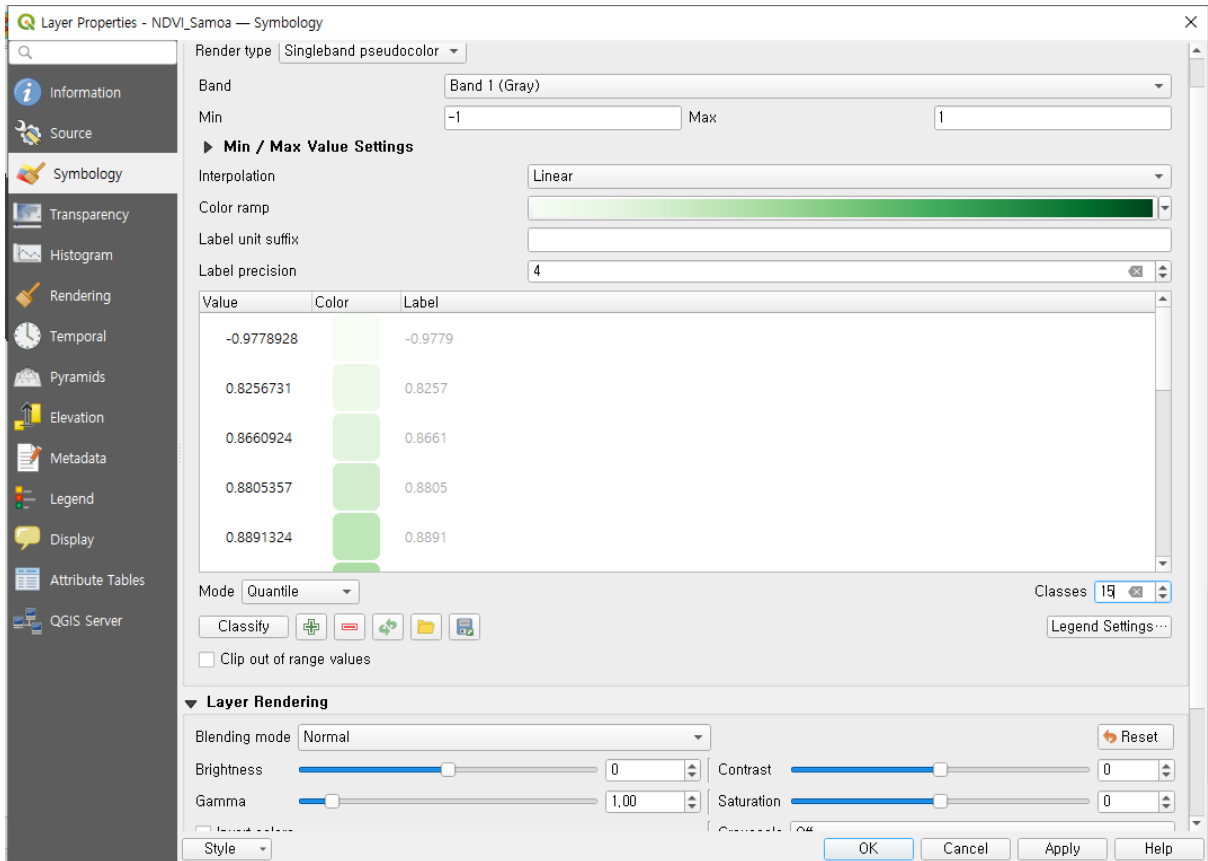
$(\text{"mosaic_13@5"} + \text{"mosaic_13@4"})$ as an Expression

- 2) Select **Practice_BGRNIR [EPSG:4326]** as a Reference layer(s), **0.000000 (Set as a default)** as a Cell size, **Use Canvas Extent** as an Output extent, and **EPSG:4326 – WGS 84** as an Output CRS

2. Run and Save as **Samoa_NDVI.tif**



3. Check the value of NDVI, which range is -1 to +1, and change the symbology to enhance vegetation area
 - 1) Select **Singleband pseudocolor** as a Render type, **Linear** as an Interpolation, **Find green color** as a Color ramp, **Quantile** as a Mode, and **15** as a Classes
4. Click **Apply > OK**



5. Compare between True color and false color images

