

# 4<sup>TH</sup> ESA EARSEL CNR SCHOOL

## DEMO: MACHINE LEARNING FOR BURNT AREA MAPPING

***Dimitris Stavrakoudis***

*PhD Electrical & Computer Engineer*

*Laboratory of Forest Management & Remote Sensing*

*Aristotle University of Thessaloniki*

*These are some notes on how to install QGIS, its plugins, and Python libraries used for exercise, as well as a description of the steps followed during the demo. For any questions, feel free to contact me directly by email at [jstavrak@auth.gr](mailto:jstavrak@auth.gr).*

### CONTENTS

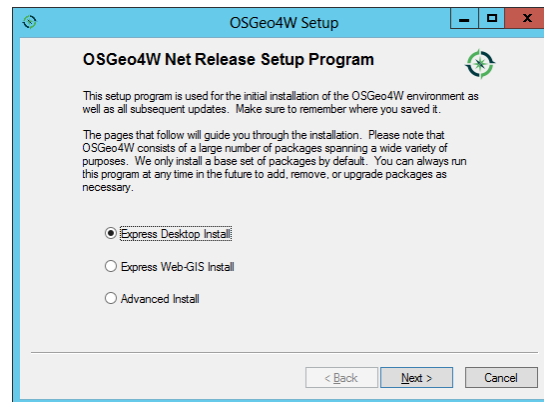
Installing QGIS and Additional Plugins/Libraries .....	2
Installing QGIS Plugins .....	4
Data Used .....	5
Sentinel-2 Image Loading and Preprocessing .....	5
Simple Approach: Threshold Spectral Indices .....	11
More Complex Approach: Supervised Learning .....	16
Combining the Two Approaches .....	20
Final Notes .....	24

## INSTALLING QGIS AND ADDITIONAL PLUGINS/LIBRARIES

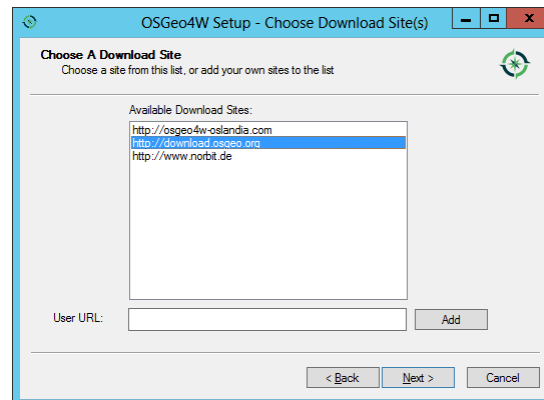
If you are using Windows, download and run the [OSGeo4W](http://download.osgeo.org/osgeo4w/osgeo4w-setup-x86_64.exe) x64 installer:

[http://download.osgeo.org/osgeo4w/osgeo4w-setup-x86\\_64.exe](http://download.osgeo.org/osgeo4w/osgeo4w-setup-x86_64.exe)

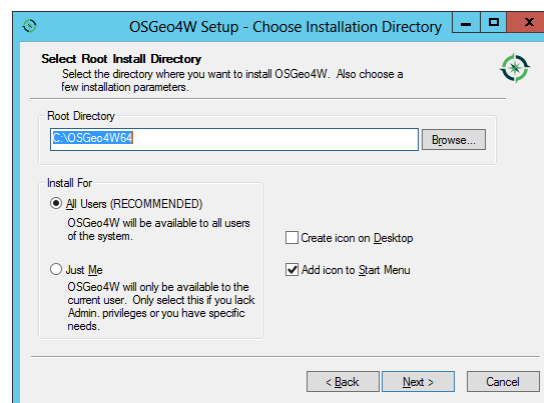
Select Express Desktop Install on the first screen and press next:



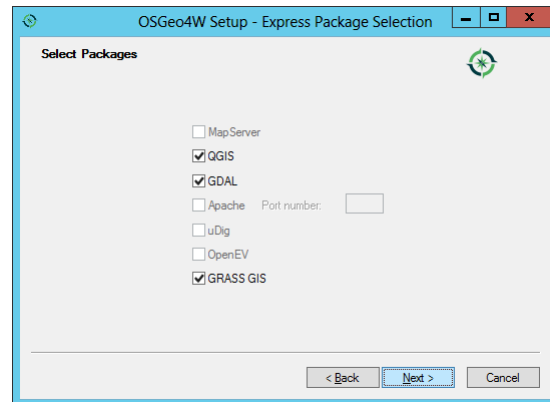
On the next screen select any download site (the second is the OSGeo official one):



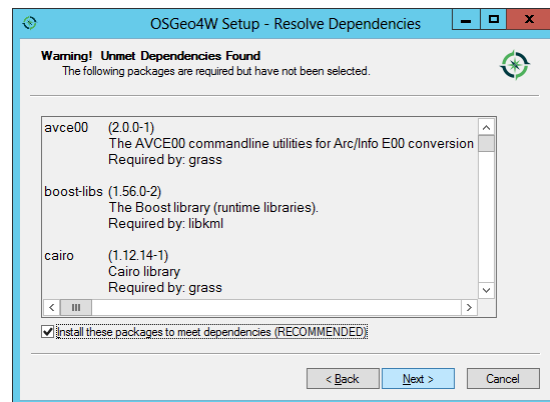
By default, all programs will be installed on C:\OSGeo4W64. If you prefer a different folder, press **Back** three times now and change it to the desired folder. Then press Next three times again to go to the Download Site screen and proceed.



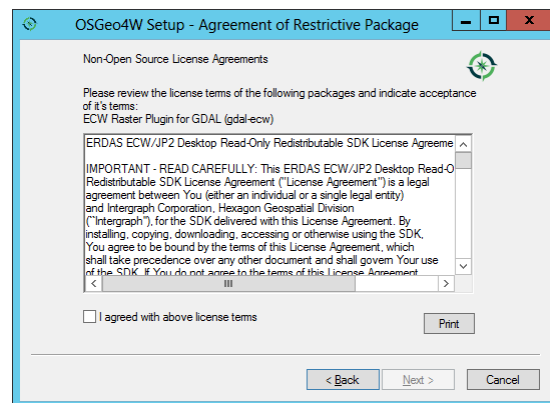
Leave QGIS, GDAL and GRASS GIS checked and proceed.



Allow the installation of all dependencies and press Next.



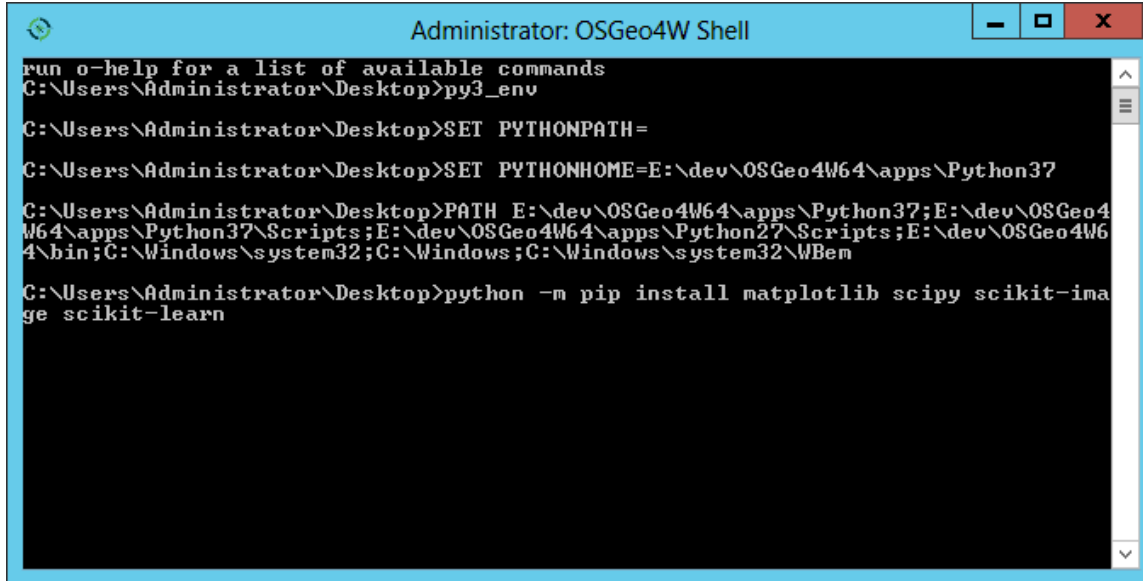
Agree to all license terms screens. These are GDAL's plugins to read proprietary image formats, such as ERDAS's ECW (JPEG2000). Wait for the installation to complete.



After installation completes, open the OSGeo4W Shell from the start menu<sup>1</sup> and type the following commands:

```
py3_env
```

```
python -m pip install matplotlib scipy scikit-learn scikit-image
```



```
Administrator: OSGeo4W Shell
run o-help for a list of available commands
C:\Users\Administrator\Desktop>py3_env
C:\Users\Administrator\Desktop>SET PYTHONPATH=
C:\Users\Administrator\Desktop>SET PYTHONHOME=E:\dev\OSGeo4W64\apps\Python37
C:\Users\Administrator\Desktop>PATH E:\dev\OSGeo4W64\apps\Python37;E:\dev\OSGeo4W64\apps\Python37\Scripts;E:\dev\OSGeo4W64\apps\Python27\Scripts;E:\dev\OSGeo4W64\bin;C:\Windows\system32;C:\Windows;C:\Windows\system32\WBem
C:\Users\Administrator\Desktop>python -m pip install matplotlib scipy scikit-image scikit-learn
```

If you are not using Windows, check the installation instruction for your OS in the official QGIS website:

<https://qgis.org/en/site/forusers/download.html>

Then use the pip command above to install the extra libraries on your system's Python 3.

## INSTALLING QGIS PLUGINS

Open QGIS and from the menu open Plugins > Manage and Install Plugins...

Search on the search bar and install the plugins **dzetsaka** and **Semi-Automatic Classification Plugin**.

---

<sup>1</sup> In some cases, the shortcut may appear within a folder OSGeo4W on your Desktop instead.

## DATA USED

We will use two Level-2A Sentinel-2 images for a wildfire event in Greece (Zemeno, Corinthia, fire started on July 23, 2018). The images can be found inside the 'data/Sentinel2\_L2A' folder on the cloud. They are for tile with ID T34SFH, the prefire one acquired on 2018-07-13 and the postfire one on 2018-07-23. You can also download them from the Copernicus Open Access Hub after registration and searching the aforementioned dates and approximately [the area around this link](#). Alternatively, the links are:

- Prefire: [https://scihub.copernicus.eu/dhus/odata/v1/Products\('f1c07754-abab-48b7-bcda-2a0e75d9daf2'\)/\\$value](https://scihub.copernicus.eu/dhus/odata/v1/Products('f1c07754-abab-48b7-bcda-2a0e75d9daf2')/$value)
- Postfire: [https://scihub.copernicus.eu/dhus/odata/v1/Products\('e0d9a9e0-6db6-4292-8485-9f27c9816c5d'\)/\\$value](https://scihub.copernicus.eu/dhus/odata/v1/Products('e0d9a9e0-6db6-4292-8485-9f27c9816c5d')/$value)

The data produced during the course can be found inside the 'data/derived\_data' folder on the cloud, but they can be produced following the instructions of this note.

## SENTINEL-2 IMAGE LOADING AND PREPROCESSING

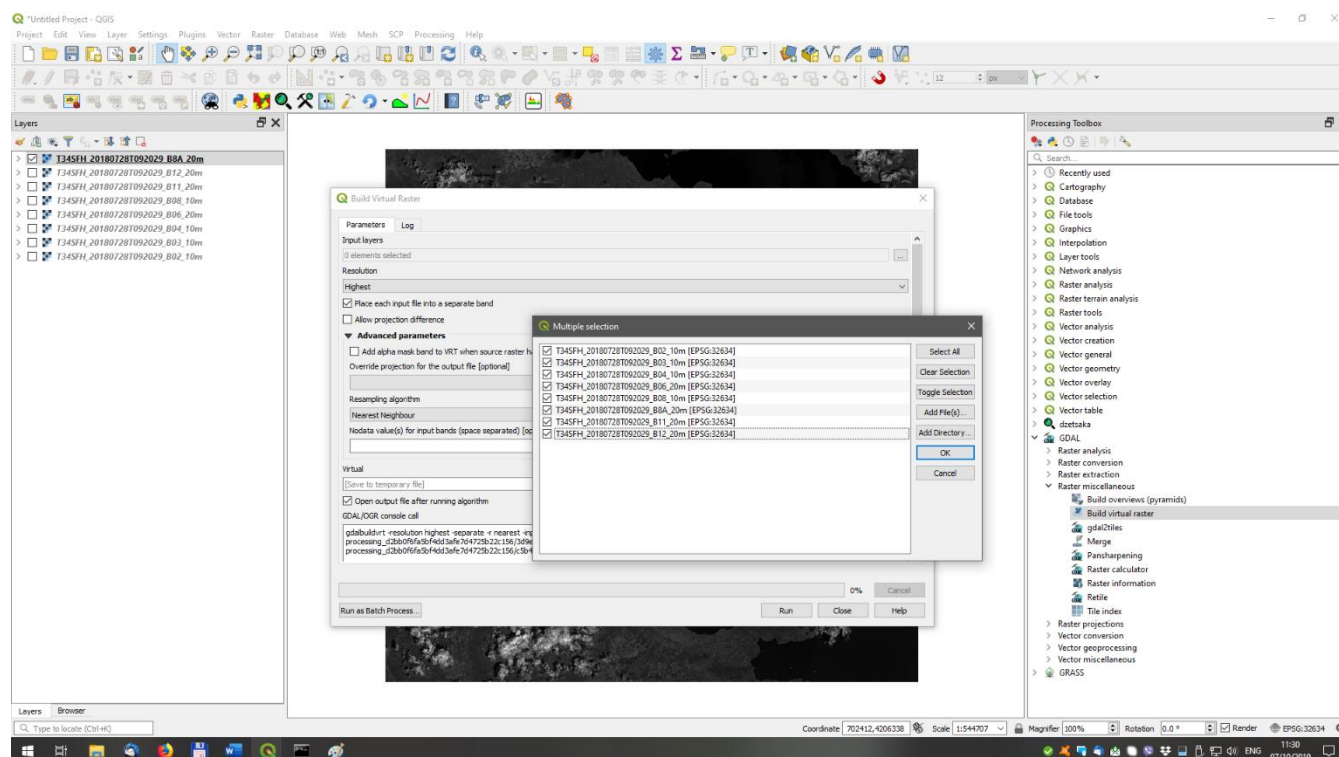
Load the postfire Sentinel-2 bands in QGIS, either through drag-and-drop from the explorer or via the 'Layer > Add Layer > Add Raster Layer...' menu. From the 'R10m' folder load bands B02, B03, B04, and B08 and from the 'R20m' folder load the B06, B8A, B11, and B12<sup>2</sup>. These are all 10 m and 20 m Sentinel-2 bands, except for extra two red-edge 20 m bands (B05: 704 nm and B07: 783 nm). The latter two are useful for calculating spectral indices related to vegetation stress, but not really useful for burned area mapping, since they do not provide any significant additional information compared to the central red-edge band B06: 740 nm that is already included. The rest of this course will use these eight bands for all processes.

Create a virtual raster<sup>3</sup>, stacking the eight individual bands into a single raster. From the Processing Toolbox on the right pane (if you don't see it, click 'Processing > Toolbox' from the menu), double left-click 'GDAL > Raster miscellaneous > Build virtual raster' (you can use the Processing Toolbox's search bar on top to locate it faster). From the Resolution dropdown list select 'Highest', in order to resample the 20m bands to 10m. Also ensure that the option 'Place each input file into a separate band' is checked, in order to stack the bands (this command can alternatively be used for creating simple mosaics if this option is unchecked). Click the three dots next to the 'Input layers' and select all eight Sentinel-2 bands. Drag and drop Band 8A (which normally will appear last) after band

<sup>2</sup> The Sentinel-2 SAFE format uses long names for individual folders, which could easily lead to a full path longer than 259 characters if you store the images inside a long base path already. On Windows, this could lead to problems and the image names may appear shortened (and incomprehensible) in QGIS. If that happens, copy the necessary bands somewhere outside (as I did on the cloud) and load them from there.

<sup>3</sup> A virtual raster is basically [an XML file following certain specifications](#), which allows to, for example, create a stacked image without copying data. If you save the file (by convention a .vrt extension is used, although not strictly necessary), you can open it with a text editor and see how this is achieved. If you save the file, the source datasets must not be deleted or change path, otherwise the VRT file won't open.

B08 and before band B11. The rest of these notes assume that the bands are stored in this order, as shown in the image below:



**Note:** The rest of these instructions assume that the band order in the stacked files is: B02, B03, B04, B06, B08, B8A, B11, B12, as shown above. This is not necessary, you can actually have any order you like, but you must remember that order and select the correct image channels for calculating the spectral indices later. If you select to have a different order, it is good to make a table like that in the following to remember the correspondence between image channels and Sentinel-2 bands.

**Note:** For each one of these Processing Toolbox commands/GUIs, there is always an option to save the file produced. By default, almost all commands have a predefined value of '[Save to temporary file]' that, as the name suggests, will create a temporary file. The 'Built virtual raster' command calls an external GDAL commandline executable and, therefore, creates an actual .vrt file, but in a temporary directory. If you close and reopen the project, this file might still exist or not, depending on the OS deleting temporary files periodically. In other cases, the temporary file might be a memory file (it will have a memory icon next to the name), which will be deleted when you close QGIS. Here, we will crop the raster in the following, so we don't care about the intermediate files, but if you want to keep it, remember to select a filename in the command's GUI (in this case, in the 'Virtual' GUI option) or by right-clicking on the name in the left pane (Layers) afterwards and selecting 'Save as'.

With the options selected above, the mapping between multi-band image channels and Sentinel-2 bands is presented in the following table, which we should keep in mind for correctly calculating the spectral indices in the following.

**Table 1** Correspondence between image channels and Sentinel-2 bands.

Image channel	Sentinel-2 band	Central Wavelength (nm)	Description
1	2	492.1	Blue
2	3	559.0	Green
3	4	664.9	Red
4	6	739.1	Red-edge
5	8	832.9	NIR (near infrared)
6	8A	864.0	NIR Narrow
7	11	1610.4	SWIR1 (shortwave infrared)
8	12	2185.7	SWIR2

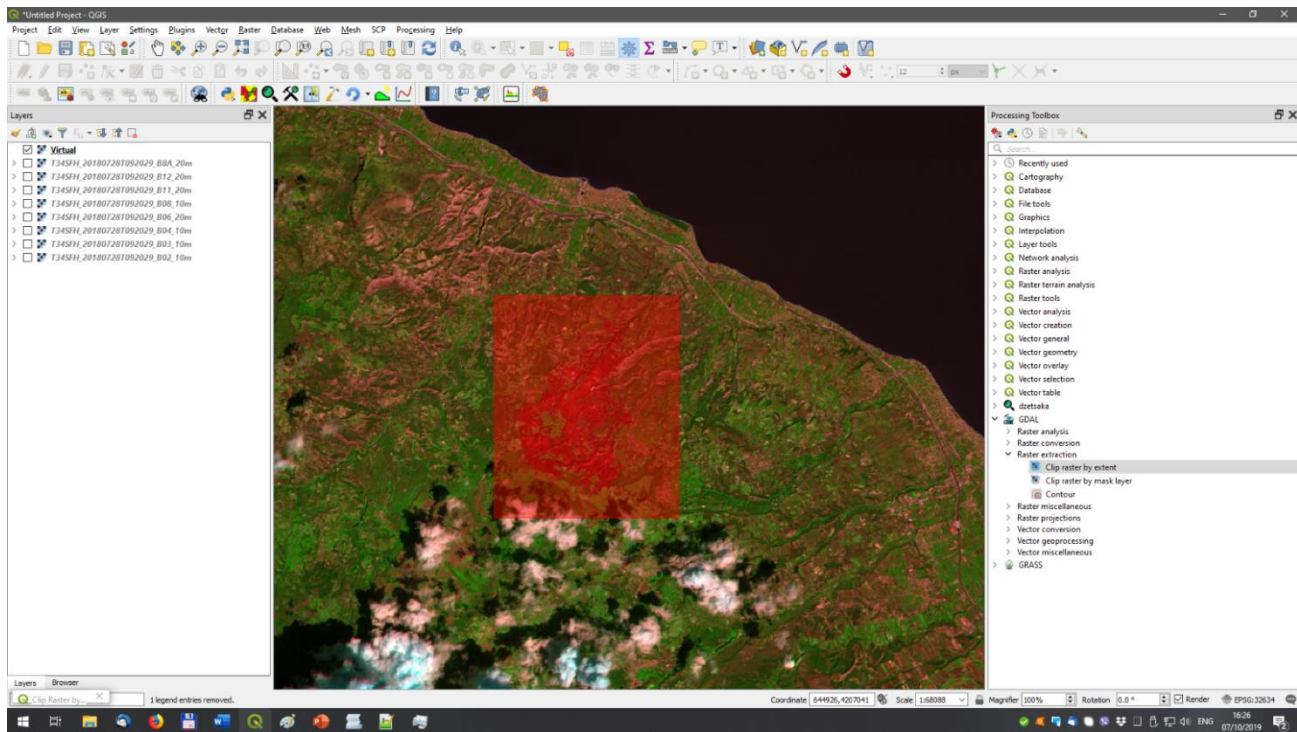
Right-click the produced 'Virtual' layer on the left pane (Layers) and select 'Properties...'. Click the 'Symbology' tab on the left and set Band 8, Band 5, and Band 3 in lieu of Red band, Green band, and Blue band, respectively. This is a common combination for viewing Sentinel-2/Landsat images (pseudo-color composite) for burned area mapping applications. From the table above, the pseudo-color composite follows the scheme below:

R ← SWIR2                      Higher values → Drier surfaces  
 G ← NIR (Band 8)              Higher values → Vegetation (and artificial areas/barren land to some extent)  
 B ← Red                          Lower values → Vegetation (but not artificial areas/barren land)

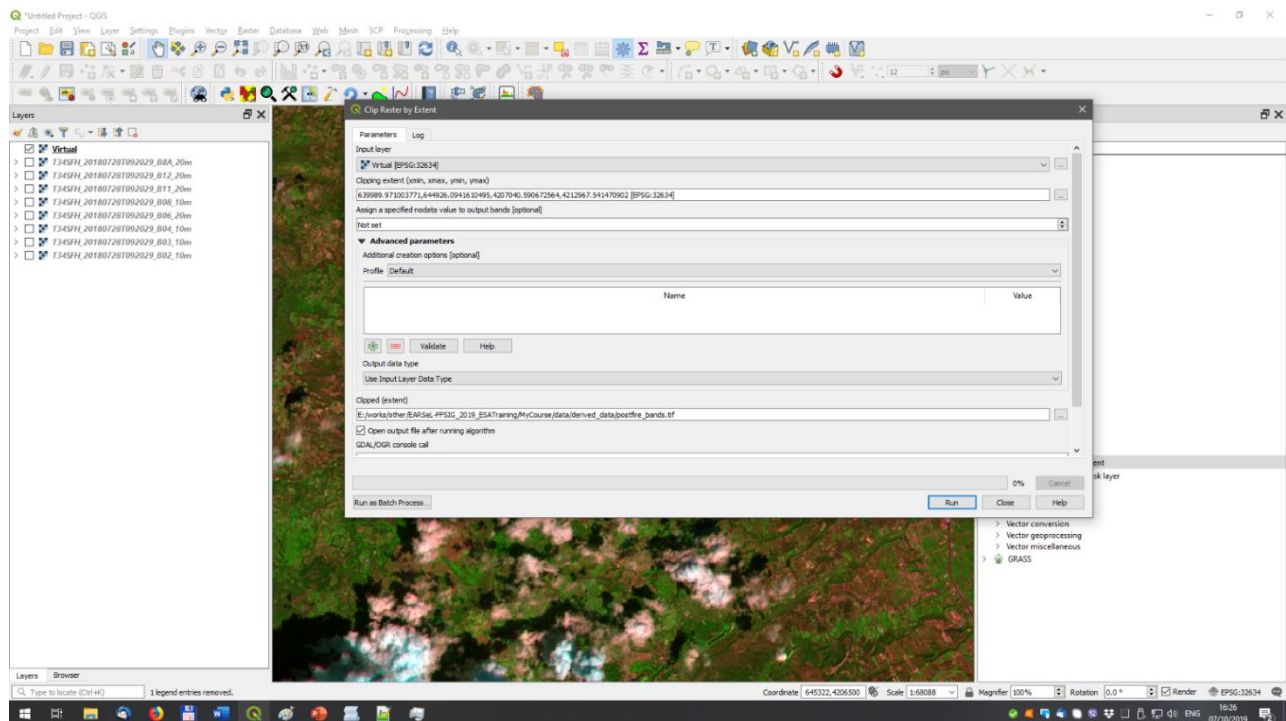
With this combination, burned areas will appear red or red/purple, healthy green vegetation as green and artificial areas/barren land as pink (think of each object's spectral properties and the RGB correspondence).

Now locate and zoom into the burned area of the example, towards the lower left part of the image. We will crop the virtual raster around the burned area. From the Processing Toolbox, open the command 'GDAL > Raster extraction > Clip raster by extent'. Make sure that the Input layer is the 'Virtual' one we created in the previous step. Click the three dots in the 'Clipping extent (xmin, xmax, ymin, ymax)' line and select 'Select Extent on Canvas'. Left-click and hold to create a rectangle that includes the burn scar (no need to be overly precise, just to include all burned area), as shown in the image below:





Finally, choose a filename to save the cropped file on your hard drive. Click the three dots next to the 'Clipped (extent)' option and select 'Save to file...'. For this tutorial, all processed data are stored in the folder 'data/derived\_data' on the ESA cloud, but you can select any folder you like on your PC. We will name the new raster 'postfire\_bands.tif' and use it throughout this tutorial (click Run to create the cropped raster):

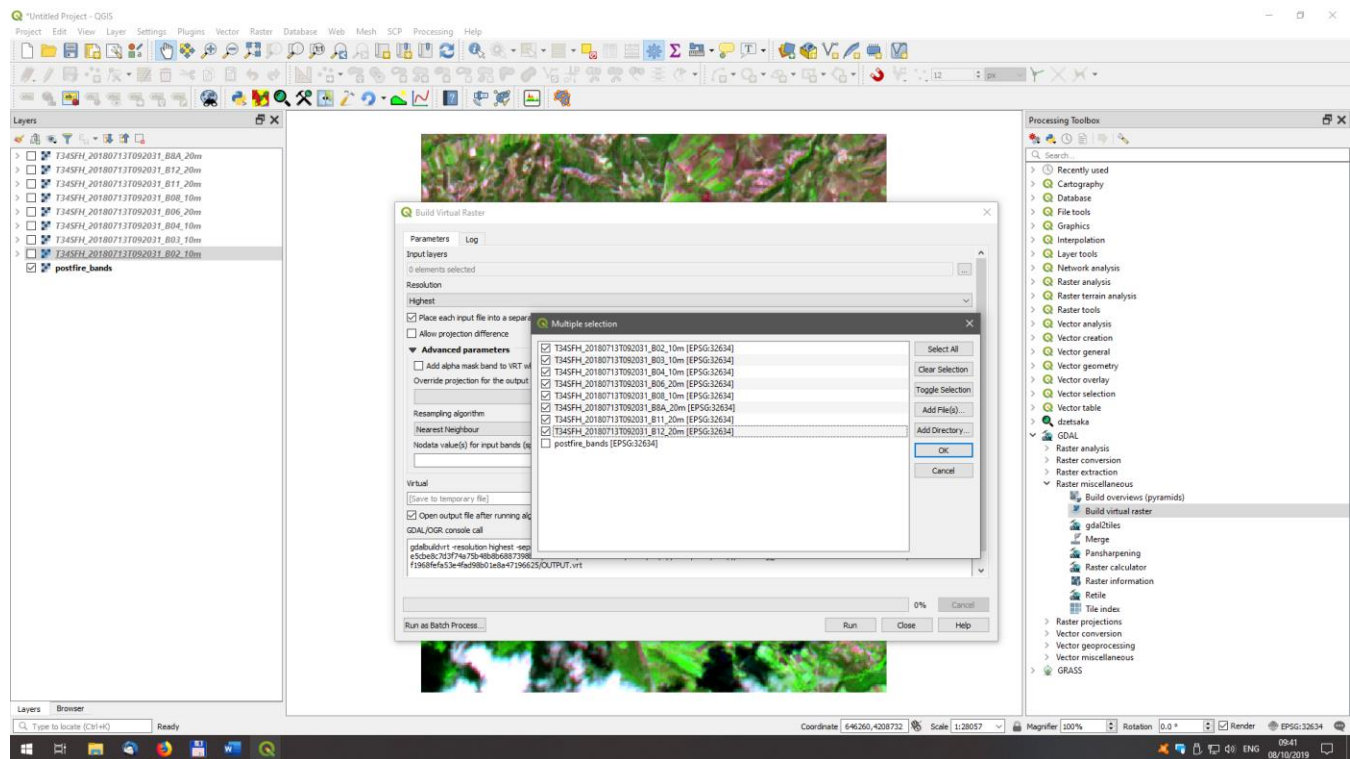




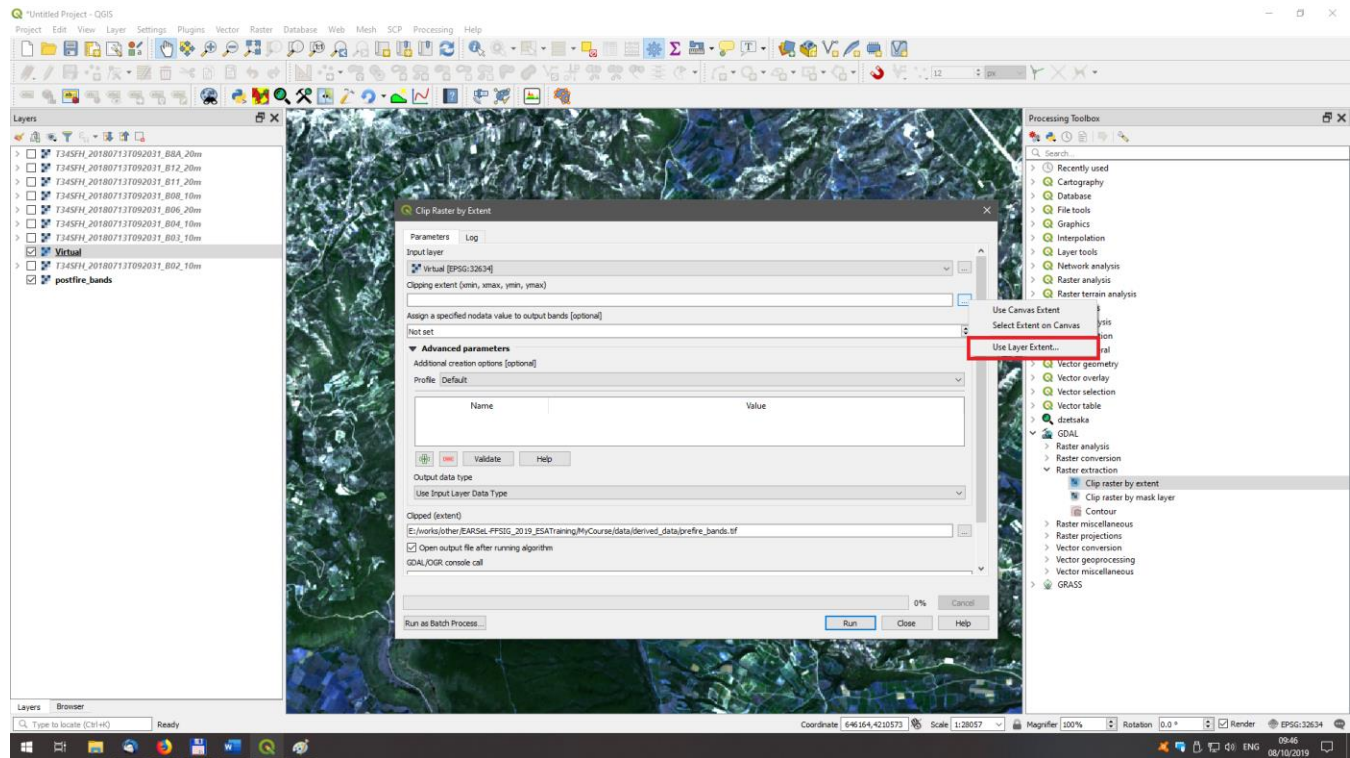
**Note:** When running Processing Toolbox algorithms, the output layer loaded in QGIS (in the Layers pane on the left) typically has a predefined name. In this case it is assigned the generic name 'Clipped (extent)'. Since we will use it in the following, rename the layer to the image name (*postfire\_bands*) or remove it and reload it by dragging and dropping the created file from your file explorer.

Right-click the new layer and from the 'Properties... > Symbology' tab set the band order as previously (R: Band 8; G: Band 5; B: Band 3).

Repeat the process above for the prefire Sentinel-2 image. First create the VRT raster (remember to move band B8A before B11 and not select the postfire cropped image!):



Then crop the image and save it as 'prefire\_bands.tif'. This time though, the extent will be set by the postfire image, choosing 'Use Layer Extent...' as the image below and then selecting the postfire\_bands layer:



That way, both images will have the same dimensions and spatial extent. Remember to rename the layer to 'prefire\_bands' in the Layers pane.

If you change the prefire symbology to the one we used so far (R: Band 8, G: Band 5, B: Band 3), you will notice that the new image's color contrast is much different than the postfire's one. To get similar color contrast between the two images, right-click the postfire\_bands layer and select 'Styles > Copy Style', then right-click the prefire\_bands layer and select 'Styles > Paste Style'. For those interested, a brief introduction to [contrast stretching can be found in this link](#).

## SIMPLE APPROACH: THRESHOLD SPECTRAL INDICES

The simplest approach to burned area mapping is to apply some threshold on specific spectral indices, using both prefire and postfire images in order to identify the change. Spectral indices are combinations of image bands through a possibly non-linear mathematical expression, which highlight properties of specific land covers. Initially, we will focus on the most well-known and used burn-sensitive spectral index, the Normalized Burn Ratio (NBR)<sup>4</sup>, which is defined as:

$$\begin{aligned} \text{NBR} &= \frac{\text{NIR} - \text{SWIR2}}{\text{NIR} + \text{SWIR2}} \\ &= \frac{\text{B8A} - \text{B12}}{\text{B8A} + \text{B12}} \\ &= \frac{\text{Channel6} - \text{Channel8}}{\text{Channel6} + \text{Channel8}} \end{aligned}$$

We use the narrow NIR Sentinel-2 band (B8A), which has a spatial resolution of 20m but its spectral response is similar to the NIR band of Landsat 8 OLI. The last line of the equation refers to the image channels of the multi-band image we have created and is only valid if the channels of the virtual raster we have created follow the order presented in Table 1.

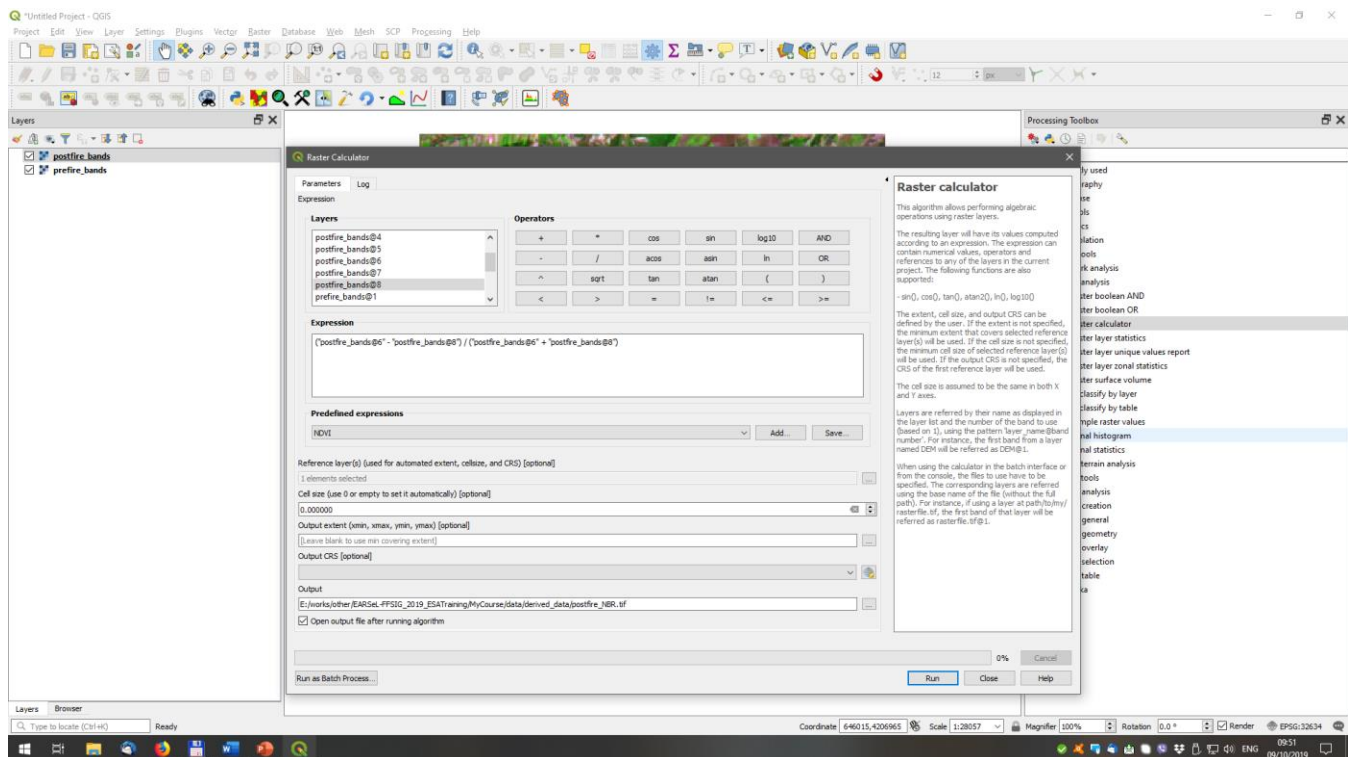
To calculate the postfire NBR, we will use the Processing Toolbox algorithm 'Raster analysis > Raster calculator'. We have to fill the expression, which follows the convention <layer\_name>@<band number> for defining raster bands. However, the Layers panel of the algorithm's GUI lists all available bands of all loaded layers and can be inserted in the expression box by double-clicking them. Fill the NBR expression (the third line of the above equation) in the Expression box:

```
("postfire_bands@6" - "postfire_bands@8") / ("postfire_bands@6" + "postfire_bands@8")
```

Although the GUI mentions it is optional, you have to select a reference layer, by clicking the three dots next to the option 'Reference layer(s) (used for automated extent, cellsize, and CRS) [optional]' and select the 'postfire\_bands' layer<sup>5</sup>. Finally, provide a filename in the Output option (here we use the filename 'postfire\_NBR.tif') and click Run (remember to rename the layer to 'postfire\_NBR' afterwards in Layers pane):

<sup>4</sup> Key, C.H., Benson, N.C., 2006. Landscape Assessment: Ground measure of severity, the Composite Burn Index; and Remote sensing of severity, the Normalized Burn Ratio (Other Government Series No. RMRS-GTR-164-CD: LA 1-51). USDA Forest Service, Rocky Mountain Research Station, Ogden, UT. <http://pubs.er.usgs.gov/publication/2002085>

<sup>5</sup> At least one reference layer is required, because the Raster calculator can work with images of different dimensions and spatial extent, resampling the image as needed and filling any non-existing areas with zero values. Alternatively, you could have set the Output extent, the cellsize, and CRS. In our case, all images have the same resolution and spatial extent, so any of our loaded layer could actually serve as reference layer.



As you can see, NBR exhibits negative values over burned areas (higher SWIR reflectance than NIR one), which seems in our case enough for identifying the burned area. However, the postfire NBR by itself can introduce false positives, sometimes over water bodies but most importantly over bare ground (which is also dry and not vegetated). Therefore, we use the so-called difference NBR index instead, which is simply defined as:

$$dNBR = NBR_{prefire} - NBR_{postfire}$$

The idea is that non-burned areas will not have changed significantly between the two image acquisitions, so unburned bare surfaces will have a dNBR value close to zero (no change). Conversely, burned areas are expected to have a positive value. dNBR is also the oldest index used for burn severity estimation. A short description of dNBR [can be found in this link](#), whereas a more thorough discussion can be found in the NBR reference provided before or the FIREMON manual<sup>6</sup>.

We use the Raster calculator to calculate the prefire NBR, which we will store in the file 'prefire\_NBR.tif'. Now the expression becomes:

("prefire\_bands@6" - "prefire\_bands@8") / ("prefire\_bands@6" + "prefire\_bands@8")

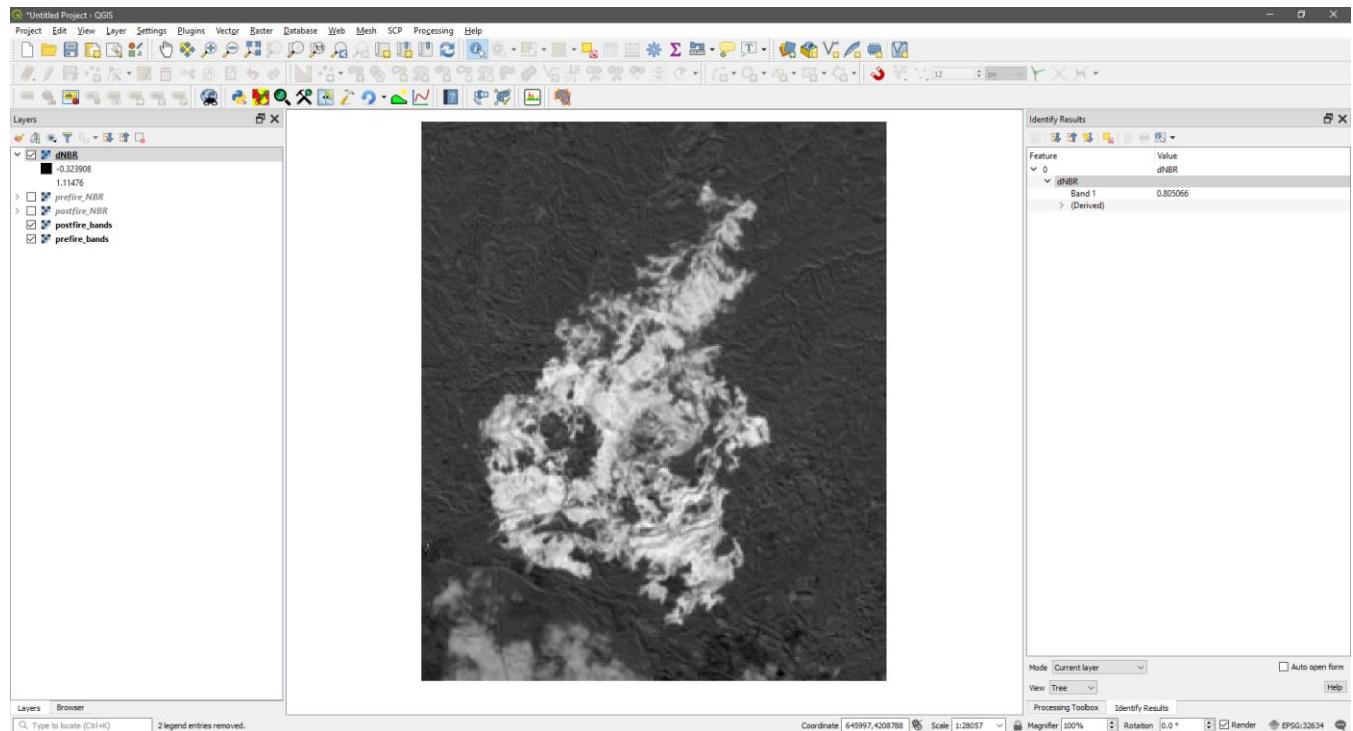
Finally, we run the Raster calculator a third time for dNBR (store it in a file 'dNBR.tif'):

"prefire\_NBR@1" - "postfire\_NBR@1"

<sup>6</sup> Lutes, D.C., Keane, R.E., Caratti, J.F., Key, C.H., Benson, N.C., Sutherland, S., Gangi, L.J., 2006. FIREMON: Fire effects monitoring and inventory system (No. Gen. Tech. Rep. RMRS-GTR-164). U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station, Fort Collins, CO. <https://doi.org/10.2737/RMRS-GTR-164>

The above expression is valid as long as you have renamed the previous layers from their default Output name to `prefire_NBR` and `postfire_NBR` in the Layers pane, as mentioned previously.

You should see this image in QGIS:



**Note:** Reflectance is a ratio, so it is defined in the range  $[0,1]$  (or 0% to 100% if you prefer). ESA provides the Sentinel-2 reflectance products (Level-1C and Level-2A) in 16-bit unsigned integer format (UInt16) to save space. UInt16 requires 2 bytes per pixel and per band, whereas a decimal (so-called floating point) format would require 4 bytes (single precision) or even 8 bytes (double precision). On top of that, UInt16 can achieve much higher compression rates in the JPEG2000 compression algorithm used. Therefore, the reflectance is multiplied by 10,000, rounded and stored in UInt16. The Raster calculator always converts the pixel values to floating point numbers before executing the expression. Because NBR is a normalized index, its calculation is not affected by the 10,000 multiplier and the result is in the range  $[-2,2]$  (the nominal one; practically it is very rare to find values below  $-1$  or above 1.5). In other cases, though (e.g., the MIRBI index, which is not normalized), you may want to divide the band by 10,000 in the Raster calculator.



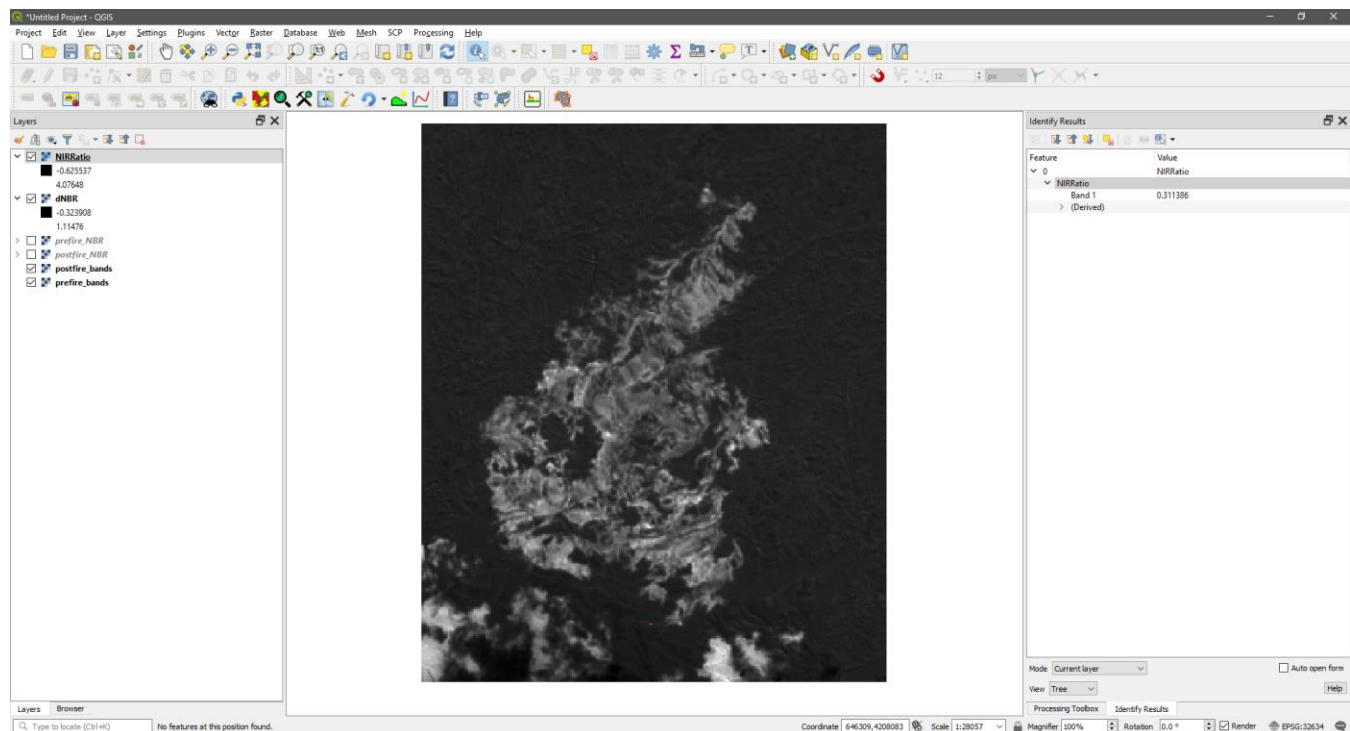
An alternative index that can be used for identifying burned areas is the  $NIR_{ratio}$  index, defined as:

$$\begin{aligned} NIR_{ratio} &= \frac{NIR_{prefire}}{NIR_{postfire}} - 1 \\ &= \frac{B8A_{prefire}}{B8A_{postfire}} - 1 \\ &= \frac{Channel6_{prefire}}{Channel6_{postfire}} - 1 \end{aligned}$$

The NIR reflectance drops after the fire over vegetated areas, so the index becomes positive in burned areas (and typically above 1, because the NIR reflectance drops at least 2 times). The  $-1$  term is used so that non-burned areas (generally no change) will exhibit values close to zero. The  $NIR_{ratio}$  index also works with satellite sensors that don't have SWIR channels, as the majority of very high-resolution commercial satellites. The index can be calculated in Raster calculator using the expression (we store it in file 'NIRRatio.tif'):

"prefire\_bands@6" / "postfire\_bands@6" -1

and you should see the following image in QGIS:



You can observe that burned areas exhibit positive values, typically over 1. Since the ratio can become even higher (over 4 in this case), you can change the image contrast from 'Properties > Symbology' (right-clicking the layer) to observe it more clearly.

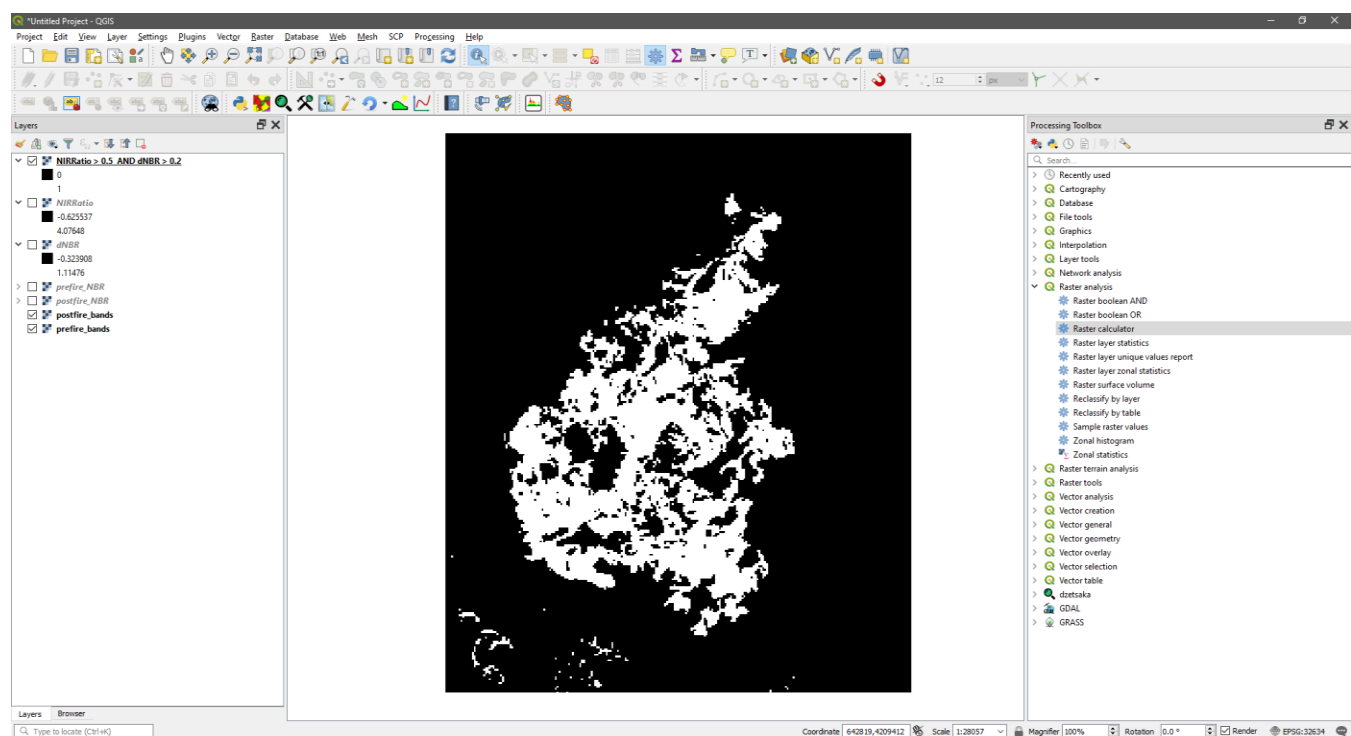
The simplest approach in burned area mapping is to apply thresholds to the index images. For example, if you check the dNBR references provided above, Key and Benson found that low-severity burns correspond to dNBR

values in the range [0.1, 0.27). Try the expression `"dNBR@1" > 0.1` and you can see that the burned area is delineated, although a number of false positives also exist; the resulting images has 1 (white color) where the relation holds and 0 (black color) otherwise. False positives can be reduced if you choose a higher threshold, e.g.,  $dNBR > 0.2$ . Accordingly, try positive thresholds for the  $NIR_{ratio}$  index (e.g.,  $NIR_{ratio} > 0.3$ ) and increase the threshold (you can leave the Raster calculator open and press Run each time; a new layer will be created in the background).

In both cases, the cloud shadows that exist in the postfire image are identified as burned areas. This is expected, because clouds actually result in changes in the pair of images and their shadows lead to a reduction in NIR reflectance. To reduce their effect, try combinations of the two indices. For example, if you try the following expression in Raster calculator:

```
"NIRRatio@1" > 0.5 AND "dNBR@1" > 0.2
```

you should get the following result (it has been stored as 'NBR\_NIRRatio\_thresholds.tif' in the 'data/derived\_data' folder on the ESA cloud):



This is a pretty good delineation of the burned area, although it has some false positives (mainly from cloud shadows) and misses some burned areas (false negative). There exist methodologies in the literature that employ only thresholds (considering more complex indices and rule sets), but in the general case all of these methodologies do present limitations as the ones shown in this example.

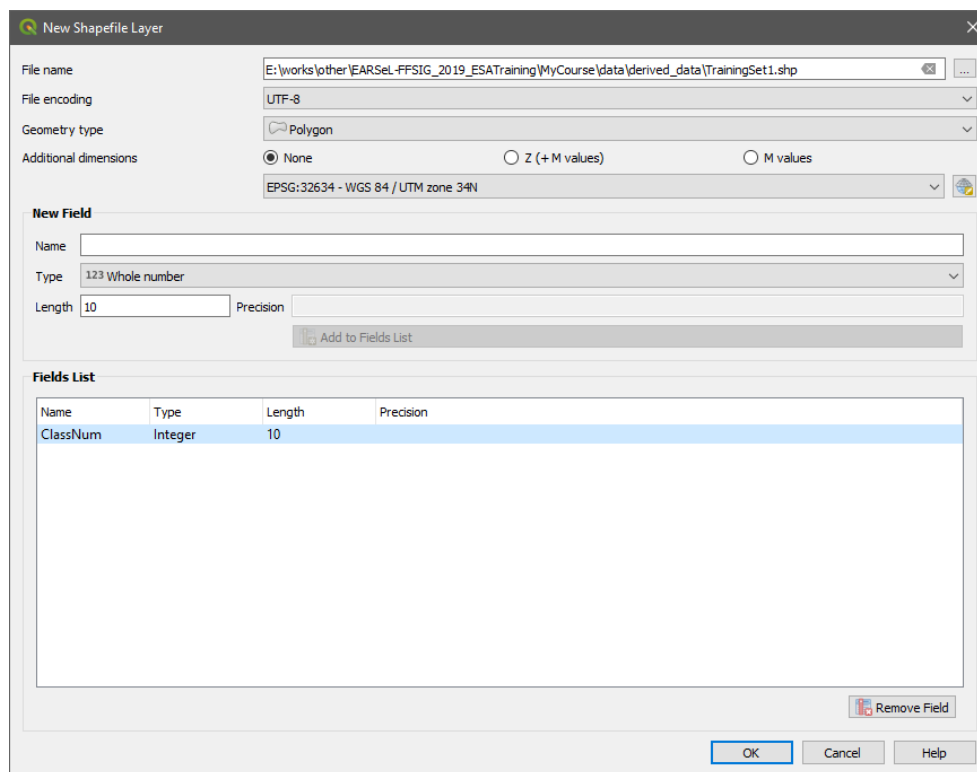


## MORE COMPLEX APPROACH: SUPERVISED LEARNING

A more systematic way to map burned areas is to follow a supervised classification approach. In supervised classification, a labeled training set is provided to a machine learning classifier, which is a statistical model with free parameters that are learned from this training set and then employed to classify the whole image. We will try such a solution, using the **dzetsaka** QGIS plugin installed earlier.

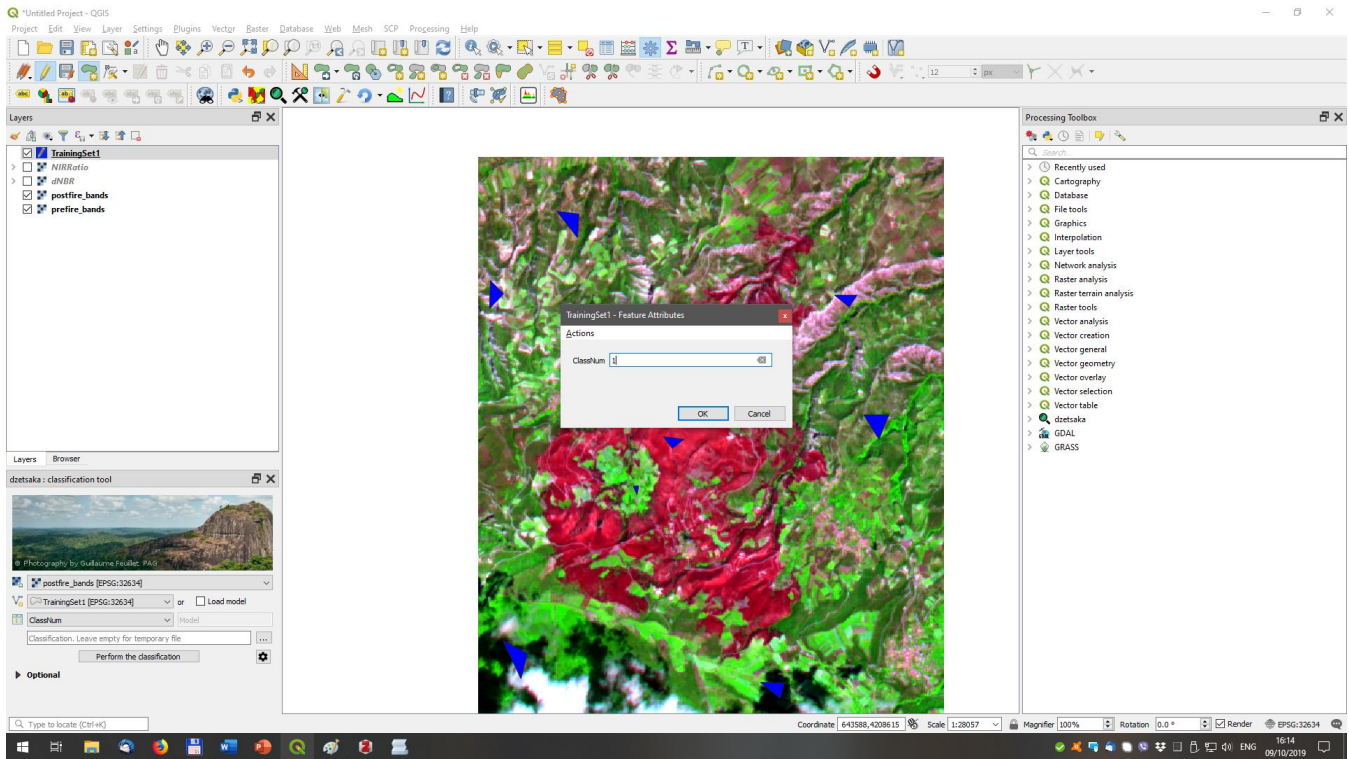
From QGIS's main menu, click 'Plugins > dzetsaka > classification dock'. Dzetsaka provides four well-known machine learning classifiers, namely, Gaussian Mixture Model (GMM), Random Forest (RF), Support Vector Machine (SVM), and K-Nearest Neighbors (kNN). GMM is gaining popularity in remote sensing the last few years and it is fast to train, so we will try that first. Select it either clicking the gear icon on the classification dock or by the main menu 'Plugins > dzetsaka > settings'.

As inputs (traditionally called features in classification), dzetsaka expects a raster layer and we will use the `postfire_bands` layer we have created. Select it on the first dropdown menu of the classification dock. For the training set, dzetsaka expects a polygon vector layer (e.g., shapefile), with an integer field (table column) that has the class number<sup>7</sup>. Here we have two classes (burned and unburned), and *the rest of this document will use the convention that **burned** is represented with **1** and **unburned** with **2***. To create the shapefile, select on the main menu 'Layer > Create Layer > New Shapefile Layer...'. Select a filename (here we use 'TrainingSet1.shp') and change the Geometry type to Polygon. Add a new field of Type 'Whole number' and Name 'ClassNum' (you can use any name actually). Remove the predefined 'id' field and click OK:

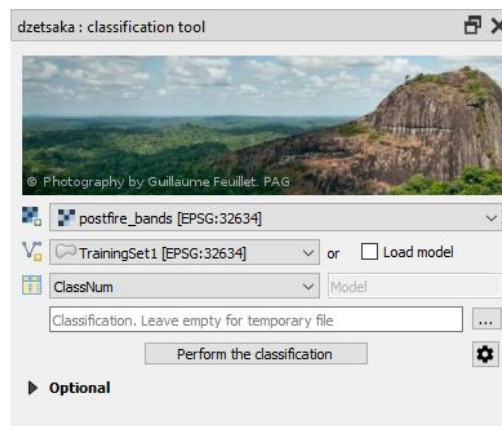


<sup>7</sup> The plugin's official page is: <https://github.com/nkarasiak/dzetsaka>

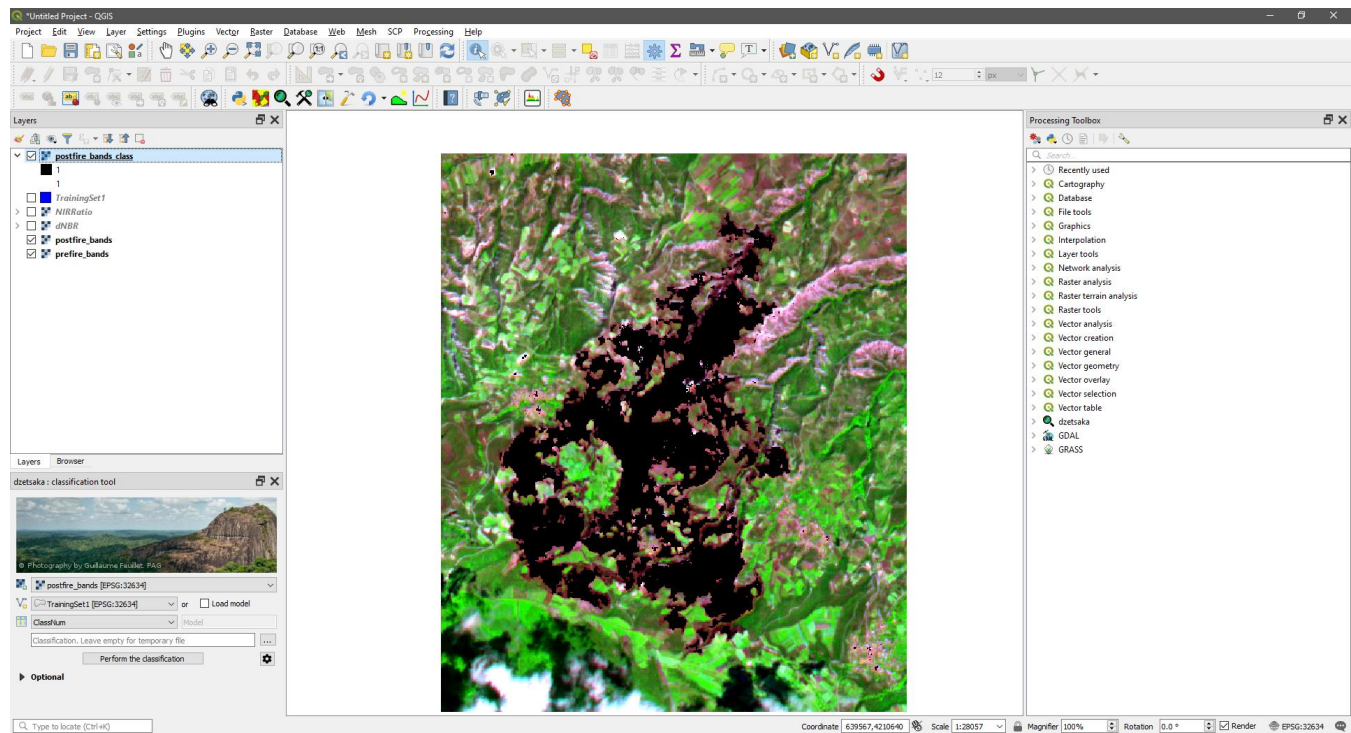
Right-click the new shapefile on the Layers pane and select 'Toggle Editing'. Press the Add Polygon button on the QGIS toolbar (📐) and start creating polygons (left click to add a new vertex, right click to end the polygon creation). No need to create complex polygons, but prefer relatively small ones because dzetsaka will rasterize them and use all pixels inside it as training, which could slow down training if they are too many. Each time you create a polygon, add its class number in the field (either 1 or 2):



When you have added some training polygons, right-click the layer and select 'Save Layer Edits' and 'Toggle Editing'. Choose the new shapefile and column in the dzetsaka classification dock (in our case it should be loaded automatically, because we have only one vector layer and it has a single integer column):

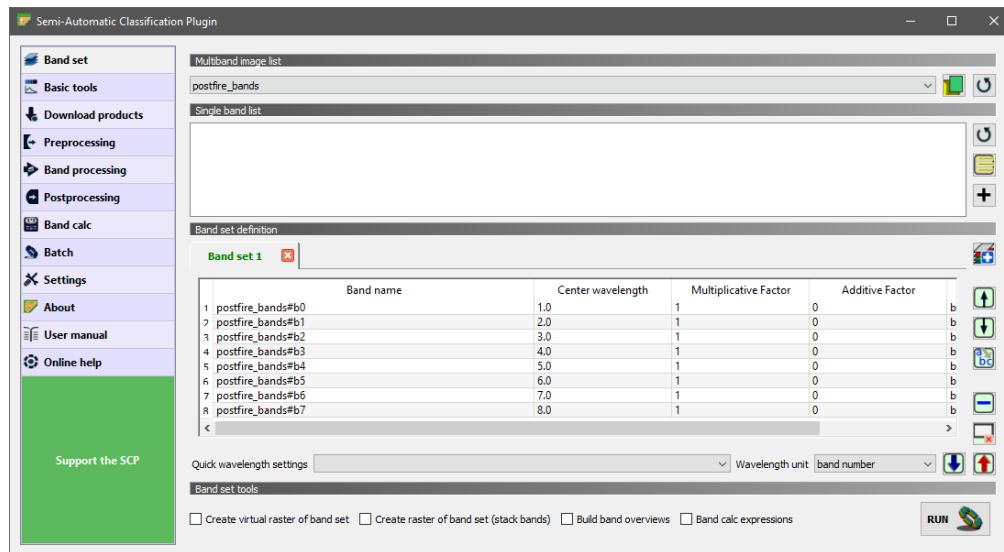


A new image is created, with 1 (black) for burned and 2 (white) for unburned. You can hide the unburned by right-clicking the output layer, and set the value 2 in 'Properties... > Transparency > No Data Value > Additional no data value'. We get something like the following image, which will be generally different if you draw your own shapefile:

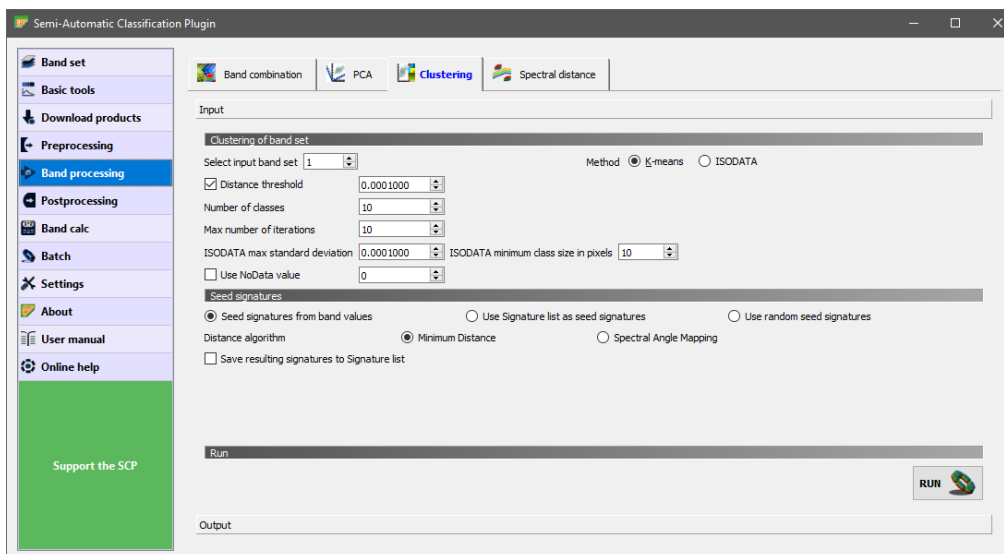


There are some false positive (classified as burned but outside the fire perimeter), as well as some omissions (classified as unburned but they are burned). This happened because our training set was not fully representative of all different land covers found in the image. To correct them, you can add additional polygons in the problematic areas in the TrainingSet1 shapefile.

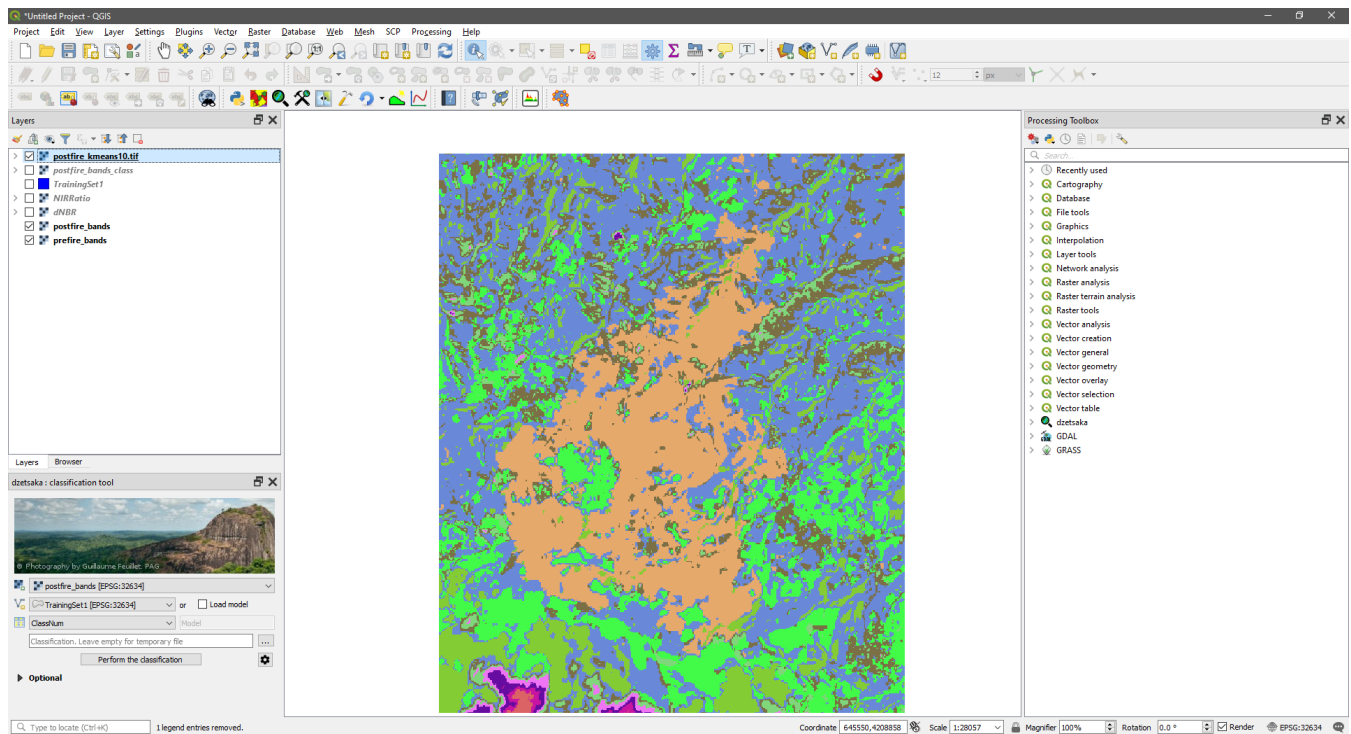
An alternative method is to gain some insights on the different categories from a clustering algorithm. Clustering is an unsupervised classification methodology, which does not require labels (training set), but tries to identify categories in the image (called clusters) with similar spectral signatures. As such, it is less accurate than supervised classification, that is, a single cluster might cover both burned and unburned surfaces with similar spectral responses, such as sparsely vegetated areas that have been burned with low severity. However, it can help in identifying those cases, by looking at the postfire image at the same type. We will use the **Semi-supervised Classification** Plugin (SCP) for performing a raster-level clustering. When installed, the plugin creates a menu item 'SCP'. Select 'SCP > Band Set' and on the 'Multiband image list' select the 'postfire\_bands' layer (if you don't see it, click the refresh button on the right of the dropdown list):



Now select the 'Band processing' tab on the left and then the 'Clustering' tab on the new window:



Leave the default options as is, which result in K-means clustering (the most well-known clustering algorithm) with 10 clusters. Click Run and save the image as 'postfire\_kmeans10.tif'. You should get a similar result as the following (colors are random each time you run it):



This image can help select the training set properly. Try to have at least two training polygons in each cluster (color) and in-between clusters of the unburned land covers. Then you can re-run the dzetsaka classification and see the new results. Also try the other classifiers offered by dzetsaka to see the differences (although they result in similar outputs most of the times; the correct training set is the most important element in the supervised classification approach).

## COMBINING THE TWO APPROACHES

We can try to combine the two approaches (threshold-based and supervised learning), in order to obtain a more systematic approach. The idea is to employ the threshold-based approach to identify all unambiguous burned or unburned pixels, which will then form the training set for a supervised classification. This time, however, we will select the thresholds so that no false positives are created. For example, we can select a threshold in dNBR that corresponds to medium to high severity. In this tutorial, the rules we will use are:

**Burned:**  $dNBR > 0.4$  AND  $NIR_{ratio} > 1.2$

**Unburned:**  $dNBR < 0.0$  AND  $NIR_{ratio} < 0.3$

The Raster calculator produces a single output raster. We want to combine both rules in a single raster, with burned assigned the value 1 and unburned the value 2, so we will use a trick to do that. The logical expressions in Raster calculator result in a raster with values of 1 if the logical expression holds and zero otherwise. So, if we multiply the second rule (unburned) by 2, we will end up with value of 2 when the rule holds and zero otherwise. If we also add the first rule, we will achieve the desired result. Ultimately, the expression we will enter is:

$1 * ("dNBR@1" > 0.4 \text{ AND } "NIRRatio@1" > 1.2) + 2 * ("dNBR@1" < 0 \text{ AND } "NIRRatio@1" < 0.3)$

Use any layer as reference and save it to a file 'TrainingSetRulesRaster.tif':

**Raster Calculator**

Parameters Log

Expression

**Layers**

- NIRRatio@1
- dNBR@1
- postfire\_bands@1
- postfire\_bands@2
- postfire\_bands@3
- postfire\_bands@4

**Operators**

+	*	cos	sin	log10	AND
-	/	acos	asin	ln	OR
^	sqrt	tan	atan	(	)
<	>	=	!=	<=	>=

**Expression**

`1*("dNBR@1" > 0.4 AND "NIRRatio@1" > 1.2) + 2*("dNBR@1" < 0 AND "NIRRatio@1" < 0.3)`

**Predefined expressions**

NDVI

Reference layer(s) (used for automated extent, cellsize, and CRS) [optional]

1 elements selected

Cell size (use 0 or empty to set it automatically) [optional]

0.000000

Output extent (xmin, xmax, ymin, ymax) [optional]

[Leave blank to use min covering extent]

Output CRS [optional]

EXTENT

Output

E:/works/other/EARSeL-FFSIG\_2019\_ESATraining/MyCourse/data/derived\_data/TrainingSetRulesRaster.tif

☒ Open output file after running algorithm

Run as Batch Process...

0% Cancel

Run Close Help

**Raster calculator**

This algorithm allows performing algebraic operations using raster layers.

The resulting layer will have its values computed according to an expression. The expression can contain numerical values, operators and references to any of the layers in the current project. The following functions are also supported:

- sin(), cos(), tan(), atan2(), ln(), log10()

The extent, cell size, and output CRS can be defined by the user. If the extent is not specified, the minimum extent that covers selected reference layer(s) will be used. If the cell size is not specified, the minimum cell size of selected reference layer(s) will be used. If the output CRS is not specified, the CRS of the first reference layer will be used.

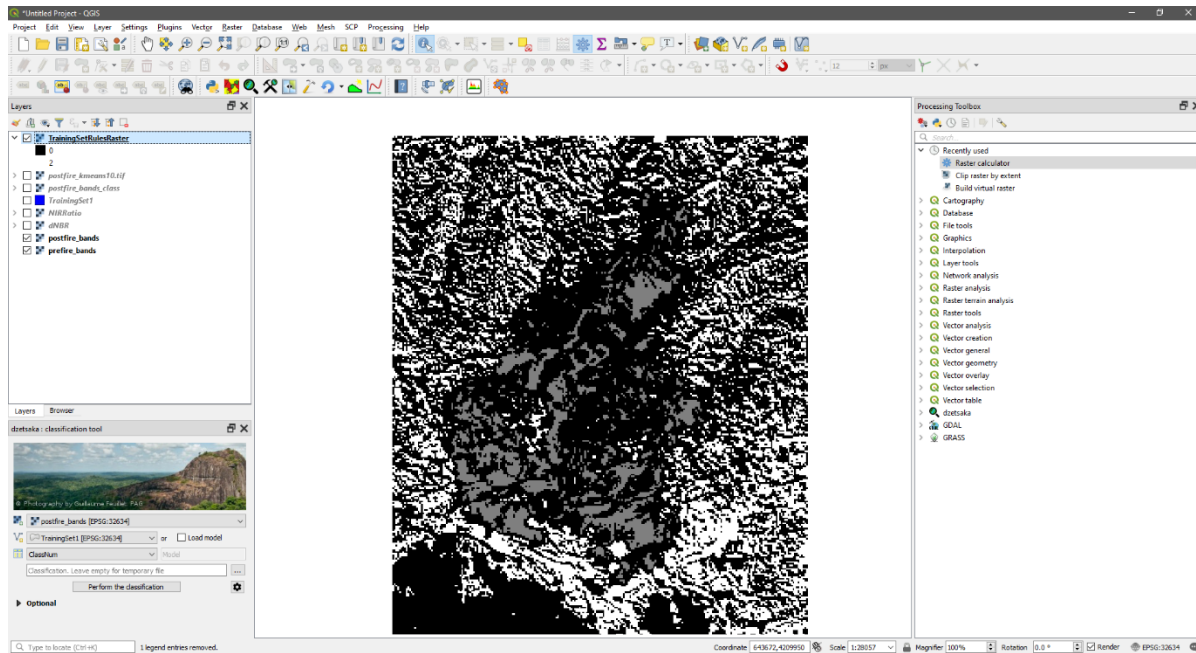
The cell size is assumed to be the same in both X and Y axes.

Layers are referred by their name as displayed in the layer list and the number of the band to use (based on 1), using the pattern 'layer\_name@band number'. For instance, the first band from a layer named DEM will be referred as DEM@1.

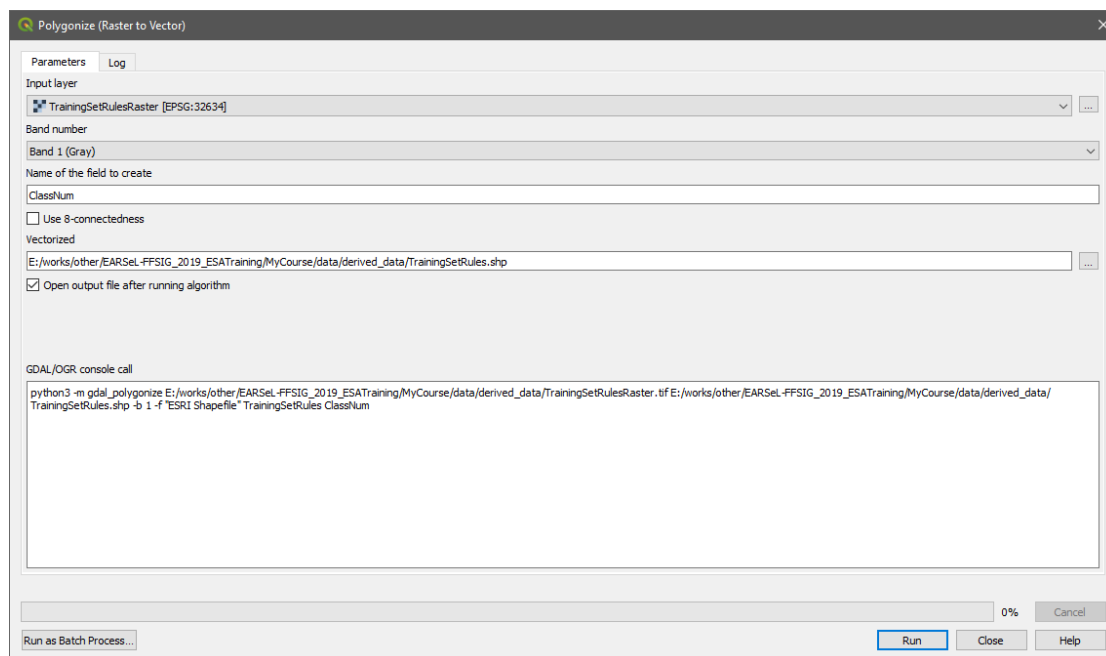
When using the calculator in the batch interface or from the console, the files to use have to be specified. The corresponding layers are referred using the base name of the file (without the full path). For instance, if using a layer at path/to/my/rasterfile.tif, the first band of that layer will be referred as rasterfile.tif@1.



We get the following image, where zero are the ambiguous pixels, which will not be inserted into the training set:

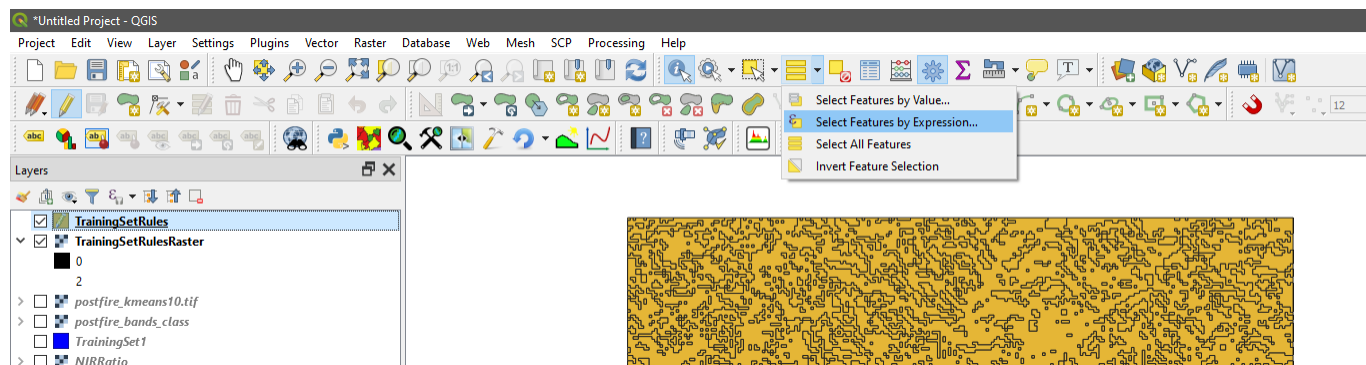


Dzetsaka expects a vector polygon layer as input, so we will convert the raster to a shapefile and delete the polygons with value 0 (neither burned nor unburned). From the Processing Toolbox select 'GDAL > Raster conversion > Polygonize (raster to vector)'. Select TrainingSetRulesRaster as 'Input' and change the 'Name of the field to create' to 'ClassNum' (although you could have let it to the default 'DN', it is just the name of the column). Save it as 'TrainingSetRules.shp' (you could work with a temporary layer as well, but you would have to save it later if you would wanted to reload the project in another time):

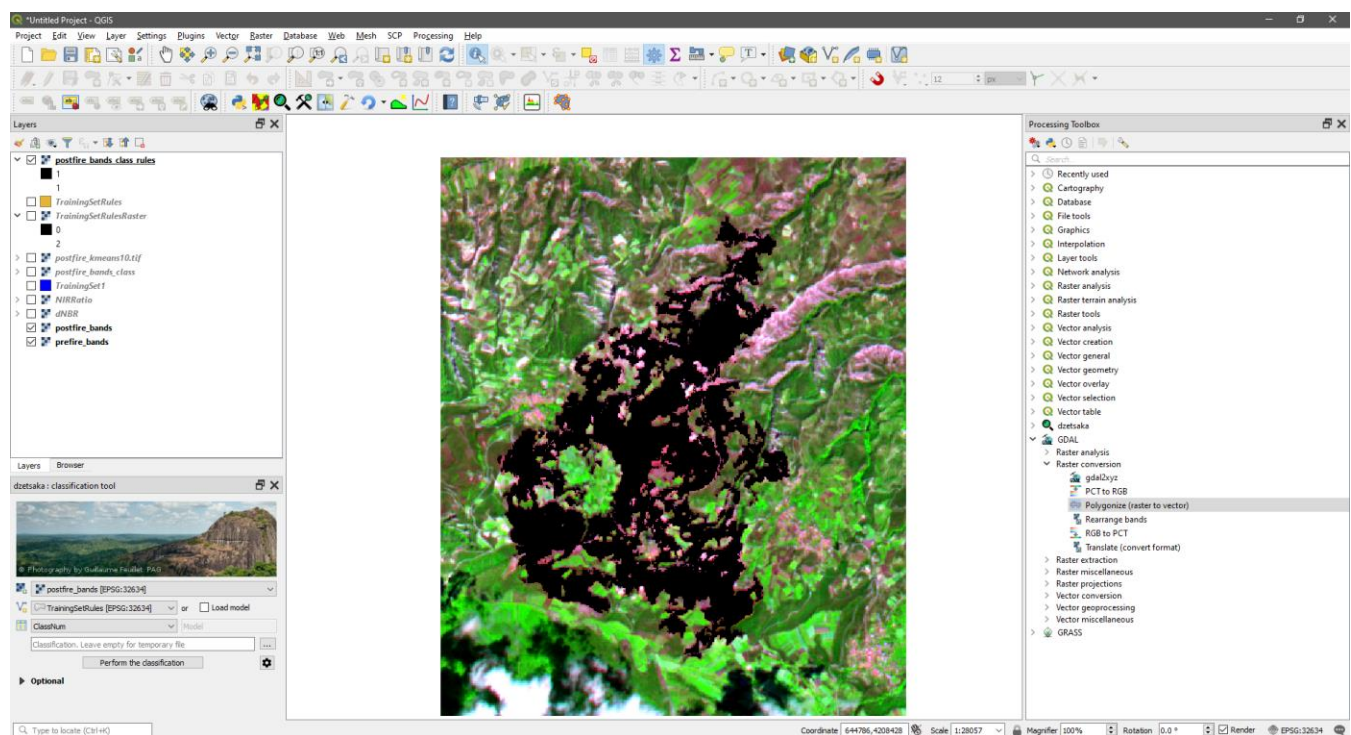




Rename the resulting layer from 'Vectorized' to 'TrainingSetRules'. Right-click the layer and select first 'Toggle Editing'. From the QGIS toolbar, choose 'Select Features by Expression...', as shown in the following image:



In the box insert `"ClassNum" = 0` and press the 'Select Features' button. This will select all polygons with a ClassNum value of 0. Now press the Delete Selected icon (🗑️) on the QGIS toolbar. Save the layer edits and toggle editing mode off. Now this shapefile can be used for training the classifier in dzetsaka as previously. The result is shown in the image below, having set the value 2 as nodata value as previously:



If you need the result simply as a burned area perimeter shapefile, you can polygonize the resulting raster classification and delete all features with value 2 (unburned).

## FINAL NOTES

The methodology presented in the previous section can be used as a basis for setting up a semi-automated burned mapping algorithm. If you are familiar with the Python programming language, a simple script could automate most of the intermediate outputs creation steps and derive the final result. Alternatively, one could use QGIS's Graphical Modeler (under 'Processing' menu item) to ease the application of commonly applied tasks. However, the empirical rules for deriving the training set are just a simplified example for this exercise. In order to work well for a reasonable number of cases, additional spectral indices should be considered. Also, the thresholds should probably be adjusted for other ecosystems.

A similar but more robust approach can be found in my presentation to the EARSeL FFSIG 2019 workshop that took place just after the 4<sup>th</sup> ESA EARSeL CNR School in Rome (October 3–5, 2019). The presentations will be shortly uploaded to the workshop's official website: <https://cnrfire2019.eu/>. This methodology follows a similar approach with that presented here, but employs an Object-based Image Analysis (OBIA) workflow, whereby the image is first segmented into objects before further analysis. This would require extensive custom programming, which is outside the scope of the current introductory course. Nevertheless, you can check it out in order to see a more realistic set of empirical rules for creating the training set automatically.