E-TRAINEE Module 2:
Satellite Multispectral Images Time Series Analysis

Theme 4: Multitemporal classification exercise

Reference dataset preparation

To perform supervised land cover classification on high-resolution, multi-spectral, and multi-temporal satellite data such as Sentinel-2 we need to prepare a reference dataset. This can be accomplished in various ways: drawing reference polygons directly from images, collecting data on the ground, or interpreting images in conjunction with other auxiliary data. In this tutorial, we'll opt for the latter approach. Reference dataset will be prepared in QGIS 3.28.2. We will use six Sentinel-2 images acquired during 2022 ("S2_2022_classification_ready.tif") and Google Satellite Images basemap to have data with higher resolution than Sentinel-2 for more accurate interpretation. Cloud masks prepared for each Sentinel-2 image were combined to prepare a new mask layer which excluded all pixels which were covered by clouds even once („mask.tif"). This ensures that our reference polygons won't overlap these cloud-covered areas. As auxiliary data we will also use the CORINE Land Cover (CLC) vector layer from 2018 (https://land.copernicus.eu/en/products/corine-land-cover), but we will verify its quality and validity. Data form the CORINE Land Cover were preprocessed for purpose of our exercise:

1. Firstly data was intersected to our Area of Interest (AoI; "T4_aoi.shp").
2. CORINE Land Cover dataset contains very detailed information about land cover, up to 44 classes for the most detailed level. For that reason classes which occur in the analyzed part of Giant Mountains were aggregated to nine classes which should be possible to classify using supervised classification (Tab. 1; "CLC_2018_Karkonosze_agg_UTM33n.gpkg").

Tab. 1 CORINE Land Cover classes aggregation scheme.

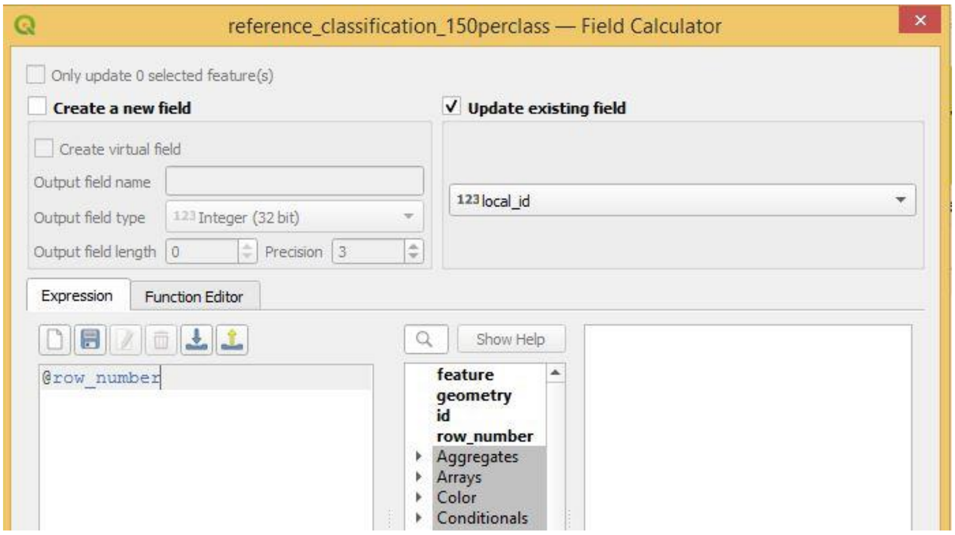
CORINE Land Cover code	CLC level 3 class name	Class name in reference data
112	Discontinuous urban fabric	built-up area
121	Industrial or commercial units	
211	Non-irrigated arable land	fields
231	Pastures	meadows
311	Broad-leaved forest	broad-leaved forest
312	Coniferous forest	coniferous forest
321	Natural grasslands	natural grasslands
322	Moors and heathland	scrub
324	Transitional woodland-shrub	
332	Bare rocks	rocks
512	Water bodies	water

Strategic partnership project funded by the Erasmus+ Program of the European Union (2020-1-CZ01-KA203-078308)

Classes 141 – Green urban areas, 142 – Sport and leisure facilities, 242 – Complex cultivation patterns, 243 – Land principally occupied by agriculture, with significant areas of natural vegetation, 313 – Mixed forest and 333 – Sparsely vegetated areas, 412 – Peat bogs were not used due to their complexity which could be an obstacle during classification. On the other hand selected nine classes contains land cover objects which are in excluded classes eg. a mixed forest contains deciduous and coniferous trees.

Since we have 6 images to classify we will use image chips to check if our polygons represent the same land cover class on each image and if clouds or shadows don't appear due to masking failure which makes our polygons unrepresentative. To prepare reference polygons several steps have been done:

1. The AoI layer was modified using a 200 m inside buffer (to not select polygons close to the image borders).
2. The regular grid of points aligned to pixel centers were generated with a 70 m gap between them (to avoid selecting polygons which are connected to each other, to reduce influence of spatial autocorrelation on accuracy assessment results).
3. Using the function „Join attributes by location” attributes „Code_18” (CORINE program code) and „Class” from CLC were added to previously generated points. Points situated on class patches which are not used were removed.
4. Around the remaining points a 14 m square shape buffer was created to obtain polygons with dimension 28x28 m which is close to 3 by 3 Sentinel-2 pixels. This size was chosen to ensure that during the classification process the algorithm will use spectral curves only from nine pixels, which centroids are located inside reference polygons. Our aim is to have quite homogenous and pure spectral signatures for each class.
5. Processed polygons were used in a stratified random sampling process, where 150 polygons for each class were selected and saved as a new vector layer named „reference_classification_150perclass”.
6. For layer „reference_classification_150perclass” a new attribute named „local_id” was added and filled using a field calculator by subsequent numbers. It was used to generate image chips in a similar way as it was introduced in Theme 2 exercise.



reference_classification_150perclass — Field Calculator

□ Only update 0 selected feature(s)

□ Create a new field

☑ Update existing field

□ Create virtual field

Output field name

Output field type: 123 Integer (32 bit)

Output field length: 0 Precision: 3

123 local_id

Expression | Function Editor

```
@row_number
```

7. Add one more attribute named „reference” where we will mark if a certain polygon will be used as a signature to classification or not.
8. Use layer „reference_classification_150perclass” to generate image chips (see Theme 2 exercise) which names will be created using attributes „local_id” and „class”. The R programming language will be needed to do this.

Unset

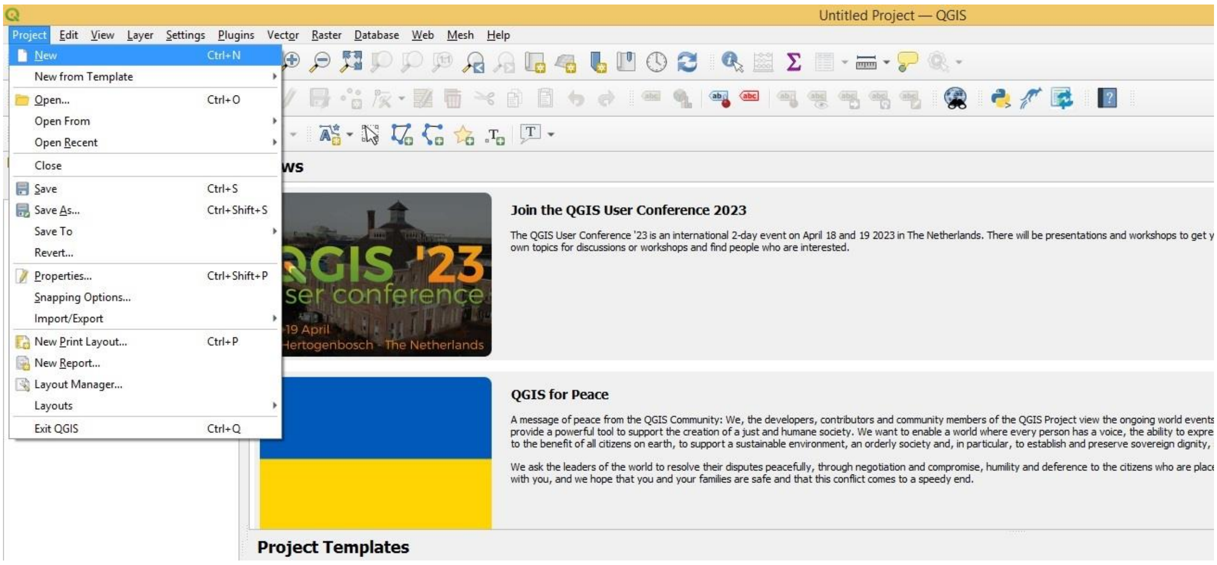
```
library(raster)
library(sf)
library(dplyr)

image <- brick("S2_2022_classification_ready.tif")

blue <- subset(image, c(1, 12, 23, 34, 45, 56))
green <- subset(image, c(2, 13, 24, 35, 46, 57))
red <- subset(image, c(3, 13, 25, 36, 47, 58))
nir <- subset(image, c(7, 18, 29, 40, 51, 62))


years <- as.Date(c( "2022-06-19", "2022-06-24", "2022-06-27", "2022-07-19",  "2022-
07-24", "2022-10-20"))

points <- st_read("reference_classification_150perclass.gpkg")

window_size <- 39

for (i in seq(nrow(points))) {


  point <- points[i, ]
  c <- st_centroid(point)

  point_cords <- st_coordinates(c)[1,]
  row <- colFromX(green, point_cords[1])
  col <- rowFromY(green, point_cords[2])

  half_widow_size <- floor(window_size / 2)
  col_cords <- (col - half_widow_size) : (col + half_widow_size)
  row_cords <- (row - half_widow_size) : (row + half_widow_size)

  output_name <- paste0("chips2/local_id == ", points$local_id[i], " ",
points$class[i], ".png")

  png(filename = output_name, width = 1920, height = 1080, pointsize = 16)

  layout(matrix(c(rep(1, 10),0, 2, 2, 0, 3, 3, 0, 4, 4, 0, 0, 5, 5, 0, 6, 6, 0, 7,
7, 0, 0, 8, 8, 0, 9, 9, 0, 10, 10, 0, 0, 11, 11, 0, 12, 12, 0, 13, 13, 0 ), nrow =
5, byrow = TRUE), heights = c(0.25, 1, 1, 1, 1))
```

Strategic partnership project funded by the Erasmus+ Program of the European Union (2020-1-CZ01-KA203-078308)

```
  par(mar = c(0, 0, 0, 0))
  plot.new()

  text(0.5, 0.5, paste0("Local ID: ", points$local_id[i], ". Class: ",
points$class[i], "."), cex = 1.4, font = 1)

  par(mar = c(0,0,1,0))



  for (j in seq(6)){

    o_b1 <- raster(matrix(getValuesBlock(green[[j]], col = row_cords[1], nrows =
window_size, row = col_cords[1], ncols = window_size),nrow = window_size, ncol =
window_size, byrow = TRUE))

    o_b2 <- raster(matrix(getValuesBlock(red[[j]],col = row_cords[1], nrows =
window_size, row = col_cords[1], ncols = window_size),nrow = window_size, ncol =
window_size, byrow = TRUE))

    o_b3 <- raster(matrix(getValuesBlock(nir[[j]], col = row_cords[1], nrows =
window_size, row = col_cords[1], ncols = window_size), nrow = window_size, ncol =
window_size, byrow = TRUE))


    b_min <- 0.1
    b_max <- 0.3
    o_b1 <- (o_b1 - b_min) / (b_max - b_min) * 255

    b_min <- 0.1
    b_max <- 0.3
    o_b2 <- (o_b2 - b_min) / (b_max - b_min) * 255

    b_min <- 0.2
    b_max <- 0.6
    o_b3 <- (o_b3 - b_min) / (b_max - b_min) * 255


    o_b1[o_b1 < 0] <- 0
    o_b2[o_b2 < 0] <- 0
    o_b3[o_b3 < 0] <- 0

    par(mar = c(0,0,1,0))
    plotRGB(brick(o_b3, o_b2, o_b1), axes = FALSE, margins = TRUE,
            xlab = "", ylab= "", main = years[j])

    symbols(x= 0.5, y=0.5, squares=0.07, inches=F, add=T, fg = "yellow")

  }
  for (j in seq(6)){

    o_b1 <- raster(matrix(getValuesBlock(green[[j]], col = row_cords[1], nrows =
window_size, row = col_cords[1],ncols = window_size),nrow = window_size, ncol =
window_size, byrow = TRUE))

    o_b2 <- raster(matrix(getValuesBlock(red[[j]], col = row_cords[1], nrows =
window_size, row = col_cords[1], ncols = window_size), nrow = window_size, ncol =
window_size, byrow = TRUE))

    o_b3 <- raster(matrix(getValuesBlock(blue[[j]], col = row_cords[1], nrows =
window_size, row = col_cords[1], ncols = window_size), nrow = window_size, ncol =
window_size, byrow = TRUE))


    b_min <- 0.1
    b_max <- 0.3
    o_b1 <- (o_b1 - b_min) / (b_max - b_min) * 255

    b_min <- 0.1
    b_max <- 0.3
    o_b2 <- (o_b2 - b_min) / (b_max - b_min) * 255

    b_min <- 0.1
    b_max <- 0.3
    o_b3 <- (o_b3 - b_min) / (b_max - b_min) * 255


    o_b1[o_b1 < 0] <- 0
    o_b2[o_b2 < 0] <- 0
    o_b3[o_b3 < 0] <- 0

    par(mar = c(0,0,1,0))
    plotRGB(brick(o_b2, o_b1, o_b3), axes = FALSE, margins = TRUE,
            xlab = "", ylab= "", main = years[j])
    symbols(x= 0.5, y=0.5, squares=0.07, inches=F, add=T, fg = "yellow")
  }

  dev.off()

  print(paste0(i, "/", nrow(points), " ", output_name))

}
```

9. Generated image chips should represent our six Sentinel-2 images in True Color and False Color (CIR) composites.

For classification purposes we want to choose 50 polygons for each class. As it was mentioned before we have randomly selected 150 polygons for each class to make it possible to choose which one met our requirements to be a good signature. Now we will create and save a new project named „Classification reference” in QGIS.



Join the QGIS User Conference 2023

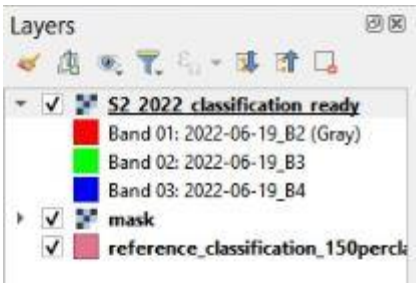
The QGIS User Conference '23 is an international 2-day event on April 18 and 19 2023 in The Netherlands. There will be presentations and workshops to get y own topics for discussions or workshops and find people who are interested.

QGIS for Peace

A message of peace from the QGIS Community: We, the developers, contributors and community members of the QGIS Project view the ongoing world events provide a powerful tool to support the creation of a just and humane society. We want to enable a world where every person has a voice, the ability to expre to the benefit of all citizens on earth, to support a sustainable environment, an orderly society and, in particular, to establish and preserve sovereign dignity,

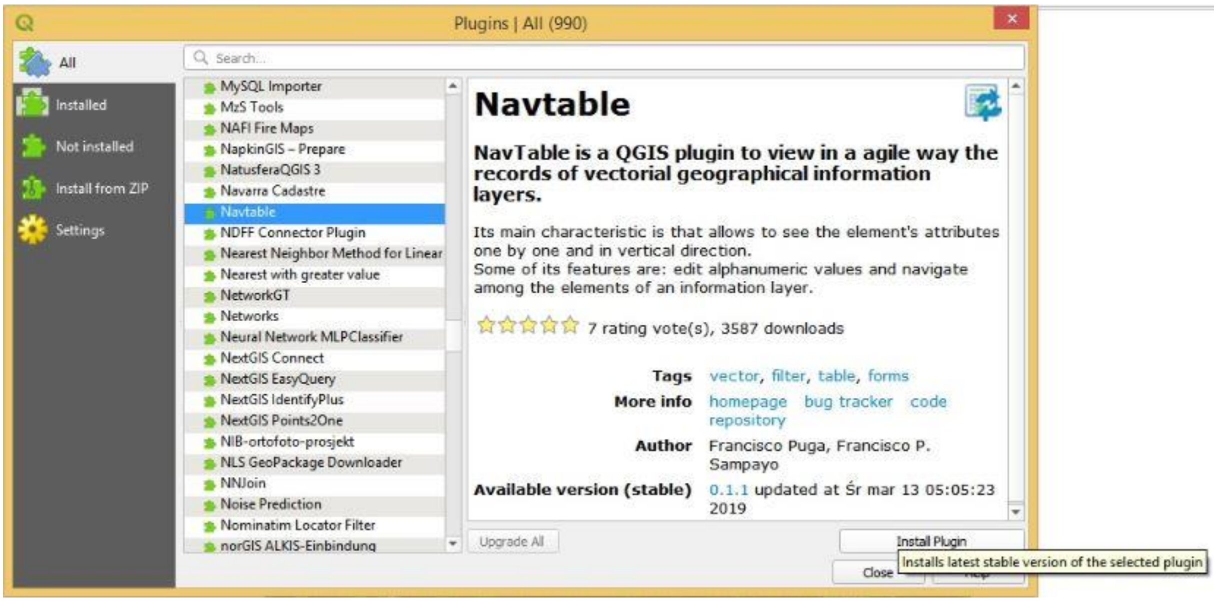
We ask the leaders of the world to resolve their disputes peacefully, through negotiation and compromise, humility and deference to the citizens who are place with you, and we hope that you and your families are safe and that this conflict comes to a speedy end.

Add the vector layer „reference_classification_150perclass” and raster layers „mask” (is combination of masks for each used Sentinel-2 images) and „S2_2022_classification_ready” (contains six Sentinel-2 images) to project.


Layers
☑ S2 2022 classification ready
🟥 Band 01: 2022-06-19_B2 (Gray)
🟩 Band 02: 2022-06-19_B3
🟦 Band 03: 2022-06-19_B4
☑ mask
☑ reference_classification_150percla

For better reference layer features management we add a plugin „Navtable”.

Strategic partnership project funded by the Erasmus+ Program of the European Union (2020-1-CZ01-KA203-078308)



Navtable

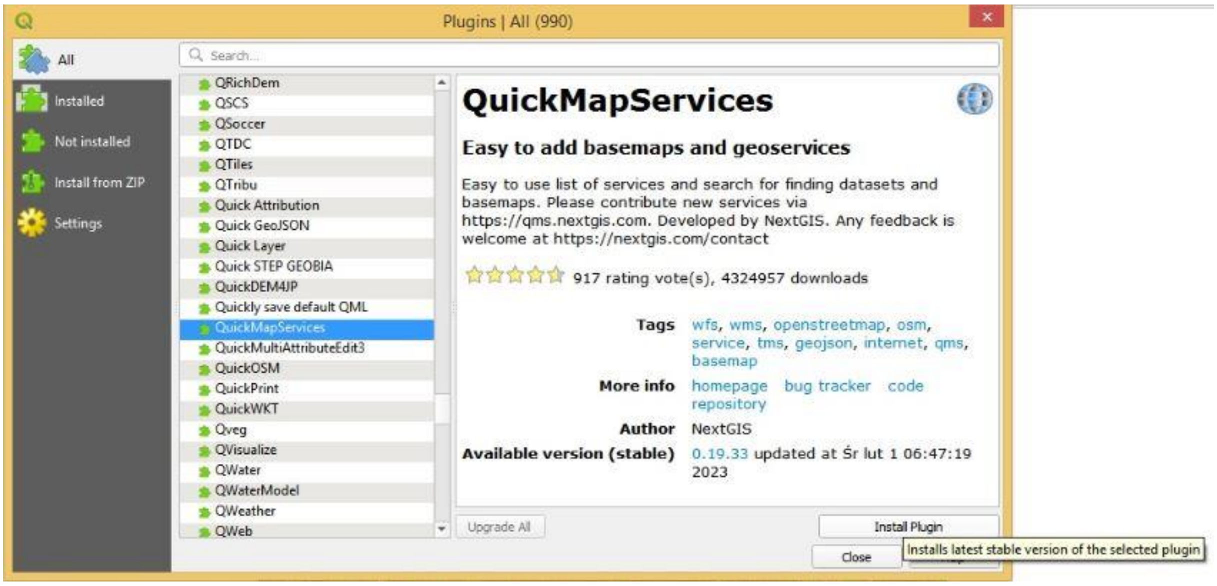
NavTable is a QGIS plugin to view in a agile way the records of vectorial geographical information layers.

Its main characteristic is that allows to see the element's attributes one by one and in vertical direction.
Some of its features are: edit alphanumeric values and navigate among the elements of an information layer.

☆☆☆☆☆ 7 rating vote(s), 3587 downloads

Tags	vector, filter, table, forms
More info	homepage bug tracker code repository
Author	Francisco Puga, Francisco P. Sampayo
Available version (stable)	0.1.1 updated at Śr mar 13 05:05:23 2019

We also add plugin QuickMapServices to have the possibility to use the Google Satellite basemap.



QuickMapServices

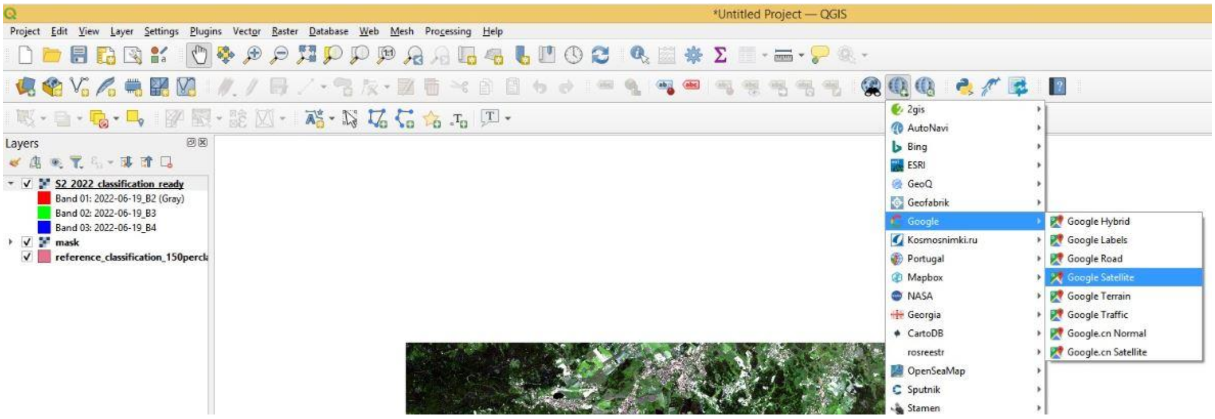
Easy to add basemaps and geoservices

Easy to use list of services and search for finding datasets and basemaps. Please contribute new services via https://qms.nextgis.com. Developed by NextGIS. Any feedback is welcome at https://nextgis.com/contact

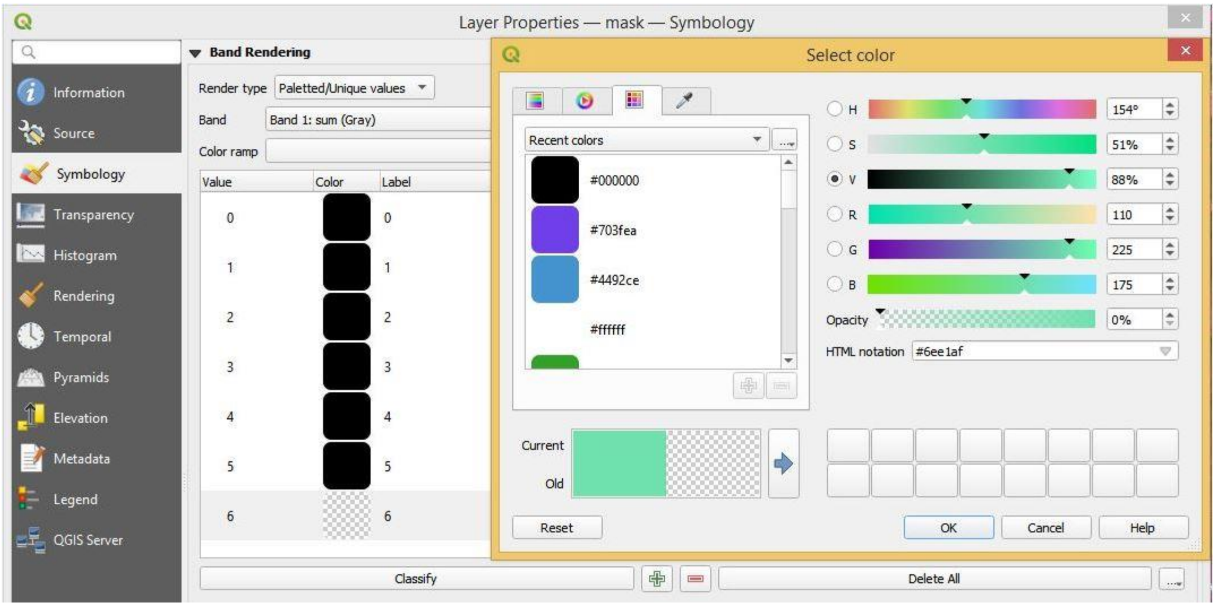
☆☆☆☆☆ 917 rating vote(s), 4324957 downloads

Tags	wfs, wms, openstreetmap, osm, service, tms, geojson, internet, qms, basemap
More info	homepage bug tracker code repository
Author	NextGIS
Available version (stable)	0.19.33 updated at Śr lut 1 06:47:19 2023

Now we will add Google Satellite basemap.



The next step is setting symbology for added layers. We will start from „mask" layer, in „symbology" tab in layer properties we should use render type as „Paletted/Unique values" and we will set black color for values from „0" to „5" (where „0" means that given pixel was masked on each image and „5" means that given pixel was masked only on one image) and we set transparent fill for value „6" (it means that given pixel was never masked).



Render type Paletted/Unique values

Band Band 1: sum (Gray)

Color ramp

Value	Color	Label
0	■	0
1	■	1
2	■	2
3	■	3
4	■	4
5	■	5
6	▨	6

#000000

#703fea

#4492ce

#ffffff

H 154°

S 51%

V 88%

R 110

G 225

B 175

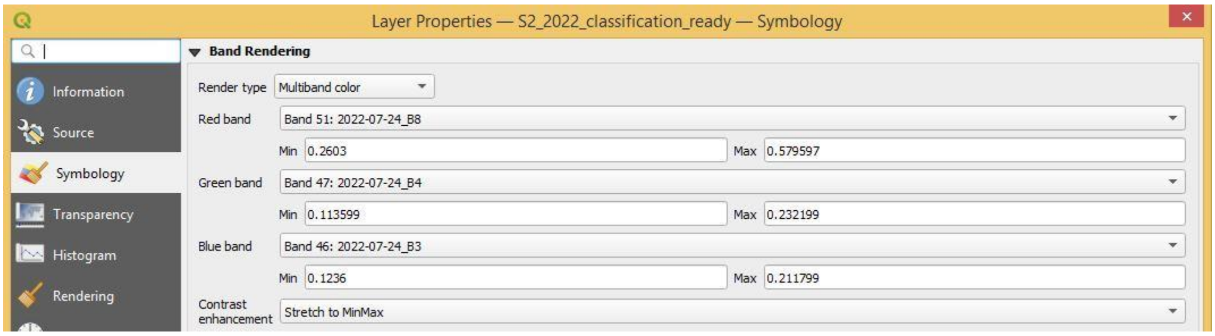
Opacity 0%

HTML notation #6ee1af

Current

Old

After setting the symbology of the „mask" layer we will go to the „S2_2022_classification_ready" layer. It contains spectral bands of six images, we will use them to visualize CIR composite for each date of acquisition.



Band Rendering

Render type: Multiband color

Red band: Band 51: 2022-07-24_B8

Min 0.2603 Max 0.579597

Green band: Band 47: 2022-07-24_B4

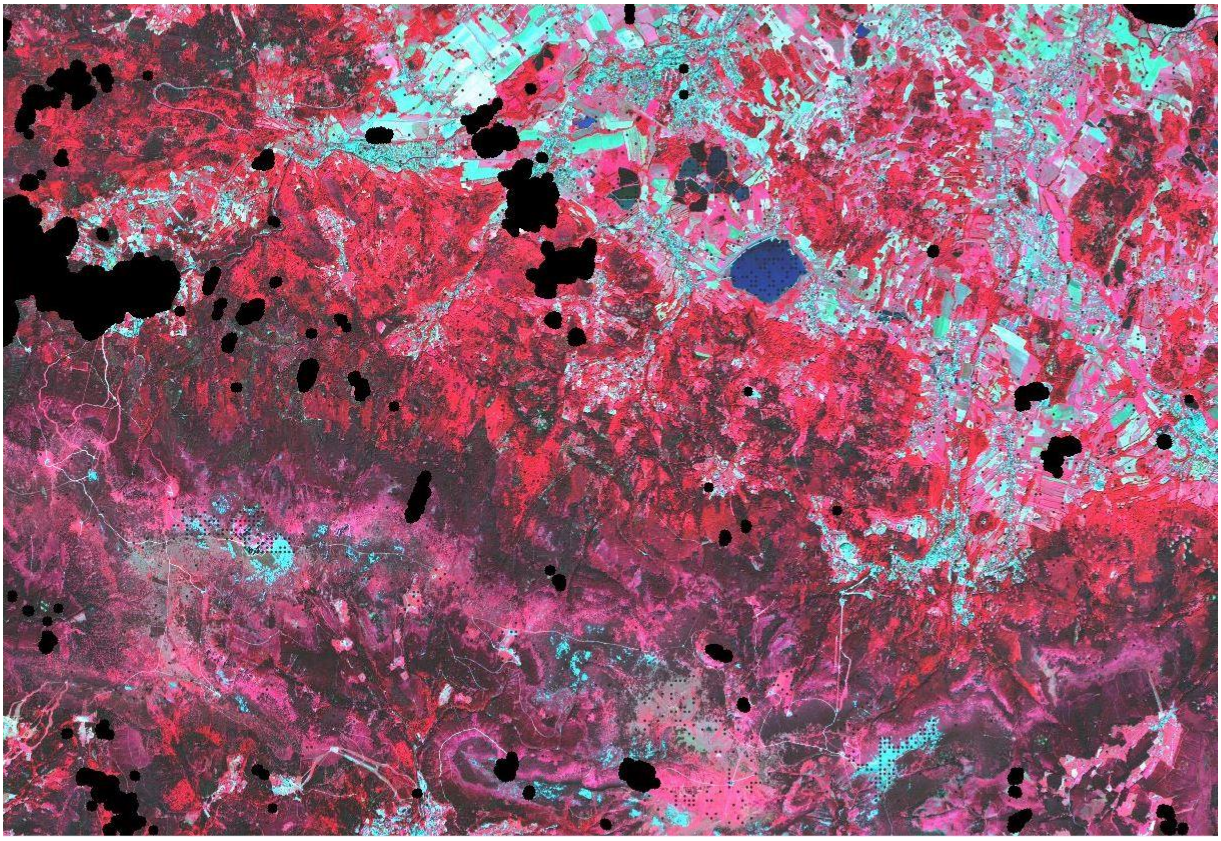
Min 0.113599 Max 0.232199

Blue band: Band 46: 2022-07-24_B3

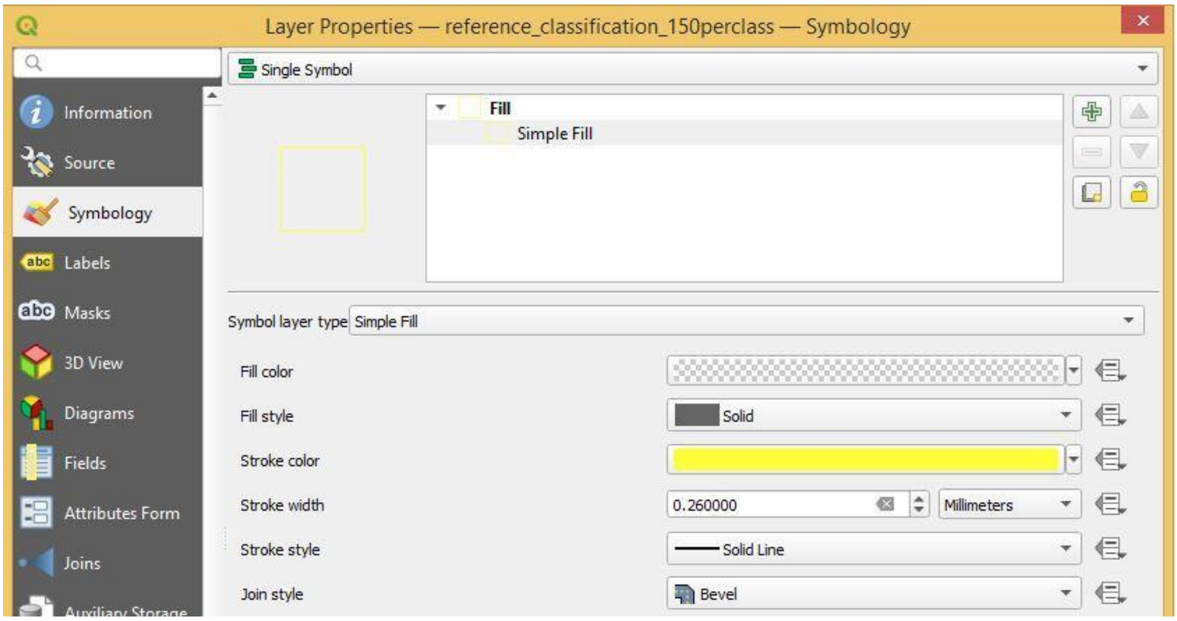
Min 0.1236 Max 0.211799

Contrast enhancement: Stretch to MinMax

Our main materials during reference dataset preparation will be image chips but composites will be helpful in some cases.



Before we will go to reference dataset interpretation we will set rules of labeling our vector layer and rules of filling values in attribute „reference”. It will make our work faster and easier. In the symbology tab in properties of the vector layer please set transparent fill and yellow color for stroke of our polygons.



Layer Properties — reference_classification_150perclass — Symbology

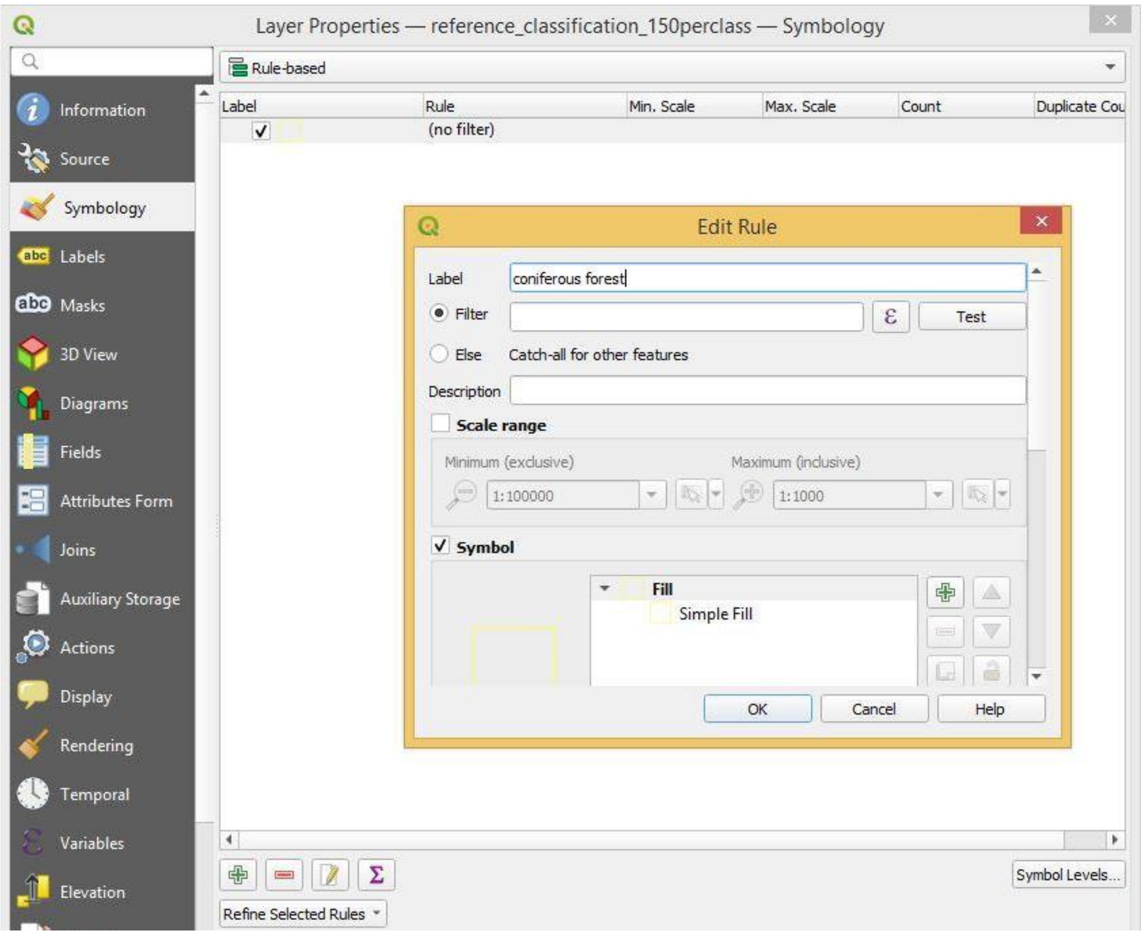
Single Symbol

Fill
Simple Fill

Symbol layer type: Simple Fill

Fill color:
Fill style: Solid
Stroke color:
Stroke width: 0.260000 Millimeters
Stroke style: Solid Line
Join style: Bevel

We will change the way of symbolization from „Single Symbol" to „Rule-based" and we will create expressions for each class. To do it we need to add each classes separately by clicking ✚ icon.



Layer Properties — reference_classification_150perclass — Symbology

Rule-based

Label	Rule	Min. Scale	Max. Scale	Count	Duplicate Cou
☑	(no filter)				

Edit Rule

Label: coniferous forest

● Filter

○ Else Catch-all for other features

Description:

□ Scale range

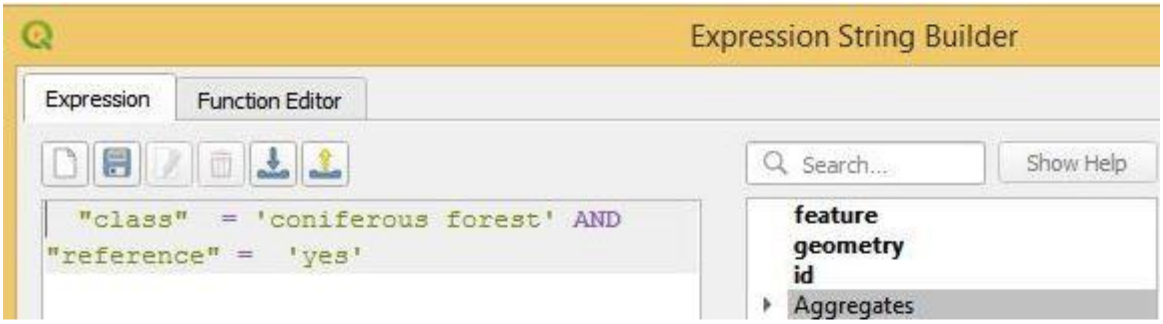
Minimum (exclusive) 1:100000 Maximum (inclusive) 1:1000

☑ Symbol

Fill
Simple Fill

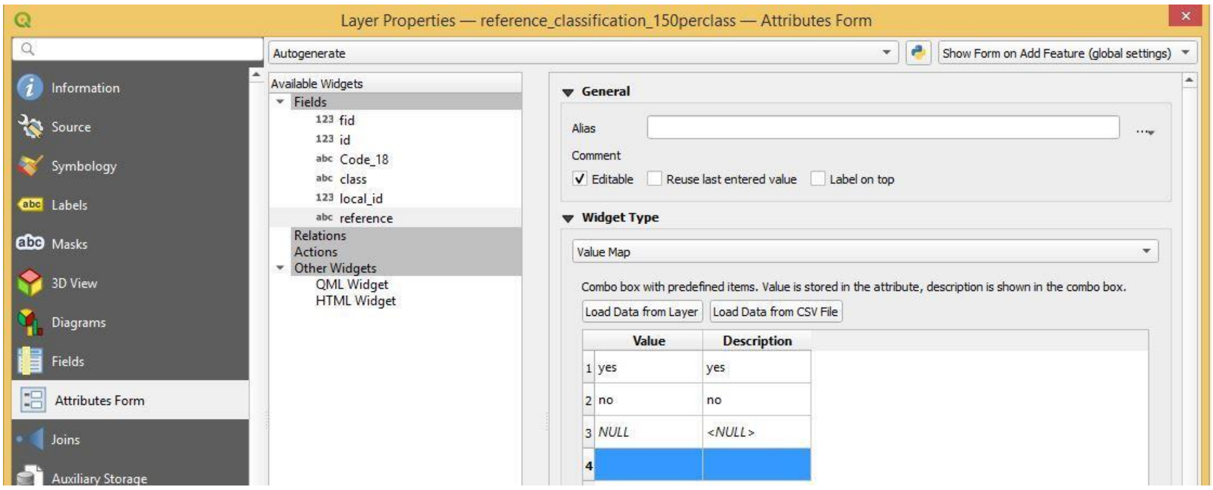
Strategic partnership project funded by the Erasmus+ Program of the European Union (2020-1-CZ01-KA203-078308)

In pop-up window we should set name of our class, change symbol choosing a green fill and black stroke and by ε clicking we can set proper expression.



```
 "class"  = 'coniferous forest' AND
"reference" =  'yes'
```

In the presented example the expression is prepared for coniferous forest class in case when attribute „**reference**” will be filled by word „yes”. To avoid filling attribute "**reference**" by typing we will add a dictionary of expression which we would use. It also allows us to avoid mistakes which can be made during typing. To prepare a dictionary we need to go to the „**Attributes Form**” tab in layer properties. For attribute „**reference**” in „**Widget Type**” section we need to select from the list option „**Value Map**”.



☑ Editable □ Reuse last entered value □ Label on top

Widget Type

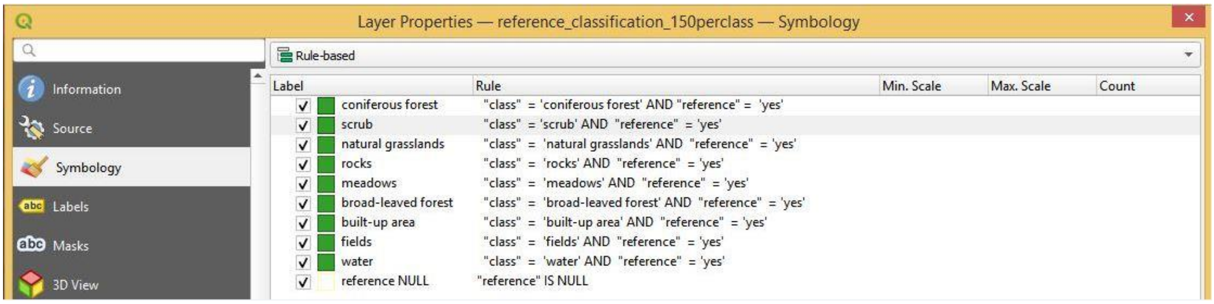
Value Map

Combo box with predefined items. Value is stored in the attribute, description is shown in the combo box.

	Value	Description
1	yes	yes
2	no	no
3	NULL	<NULL>
4		

Now we can add the desirable expressions: "yes" – means that the given polygon can be used as a reference in the classification, "no" – means that the given polygon cannot be used as a reference in the classification, "NULL" – means that the given polygon has not been verified yet. After making these settings when we will edit our vector layer to fill the attribute „**reference**” we will select those expressions from the list. In the next step, we should set expressions and symbols for the rest of classes.

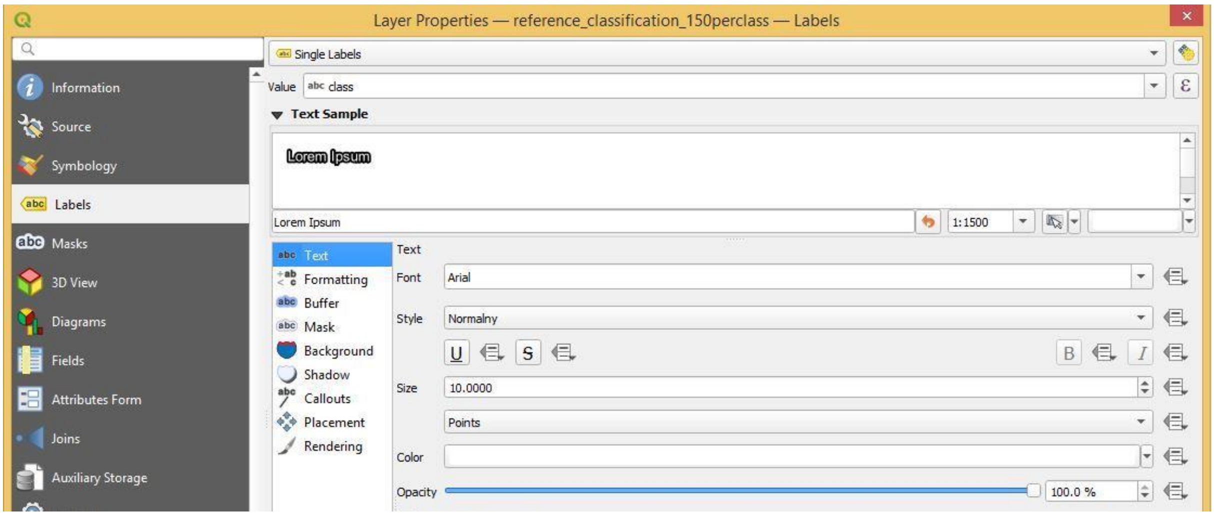
Strategic partnership project funded by the Erasmus+ Program of the European Union (2020-1-CZ01-KA203-078308)


Layer Properties — reference_classification_150perclass — Symbology
Information
Source
Symbology
Labels
Masks
3D View

Rule-based

Label	Rule	Min. Scale	Max. Scale	Count
☑ 🟩 coniferous forest	"class" = 'coniferous forest' AND "reference" = 'yes'			
☑ 🟩 scrub	"class" = 'scrub' AND "reference" = 'yes'			
☑ 🟩 natural grasslands	"class" = 'natural grasslands' AND "reference" = 'yes'			
☑ 🟩 rocks	"class" = 'rocks' AND "reference" = 'yes'			
☑ 🟩 meadows	"class" = 'meadows' AND "reference" = 'yes'			
☑ 🟩 broad-leaved forest	"class" = 'broad-leaved forest' AND "reference" = 'yes'			
☑ 🟩 built-up area	"class" = 'built-up area' AND "reference" = 'yes'			
☑ 🟩 fields	"class" = 'fields' AND "reference" = 'yes'			
☑ 🟩 water	"class" = 'water' AND "reference" = 'yes'			
☑ reference NULL	"reference" IS NULL			

Now we will go to the „**Labels**" tab to fix settings of labeling. Labels will be useful during polygon verification when a few polygons will be close to each other.



Single Labels

Value: abc class

Lorem Ipsum 1:1500

Text

Font: Arial

Style: Normalny

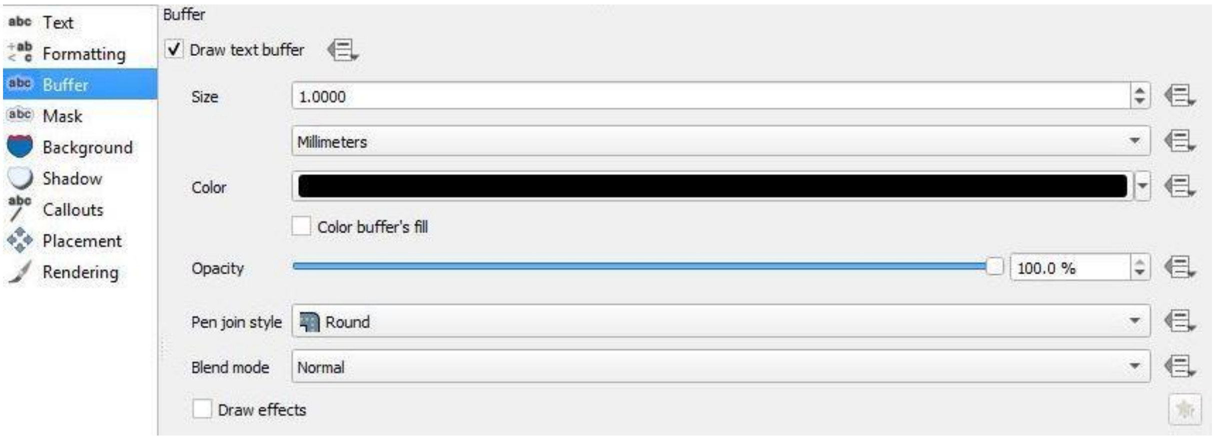
Size: 10.0000

Points

Color

Opacity: 100.0 %

We will use buffers around labels.



Buffer

☑ Draw text buffer

Size: 1.0000

Millimeters

Color

□ Color buffer's fill

Opacity: 100.0 %

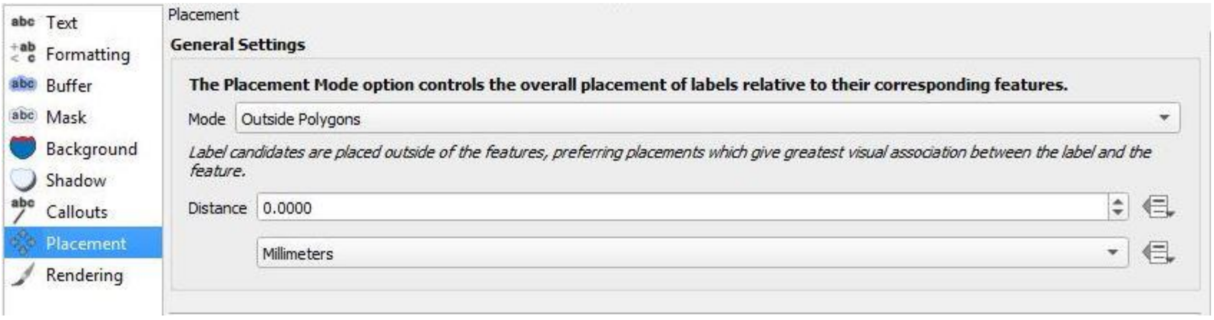
Pen join style: Round

Blend mode: Normal

□ Draw effects

Last thing which we need to set is a label placement.

Strategic partnership project funded by the Erasmus+ Program of the European Union (2020-1-CZ01-KA203-078308)



Placement

General Settings

The Placement Mode option controls the overall placement of labels relative to their corresponding features.

Mode: Outside Polygons

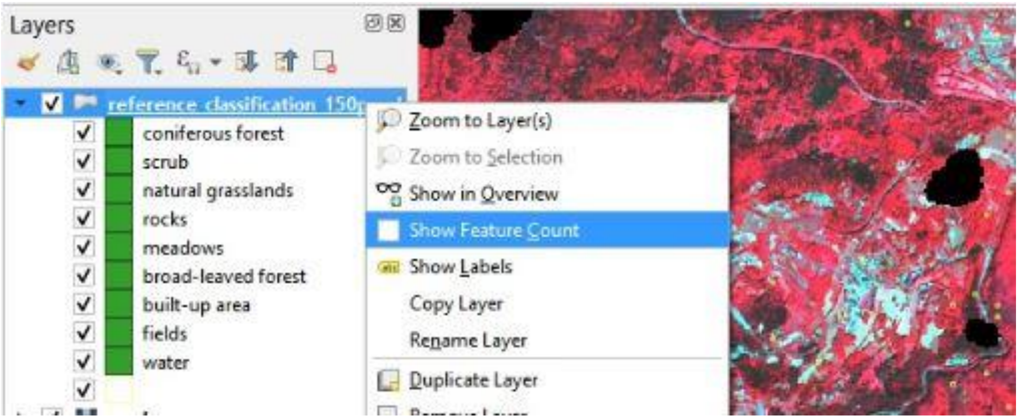
Label candidates are placed outside of the features, preferring placements which give greatest visual association between the label and the feature.

Distance: 0.0000

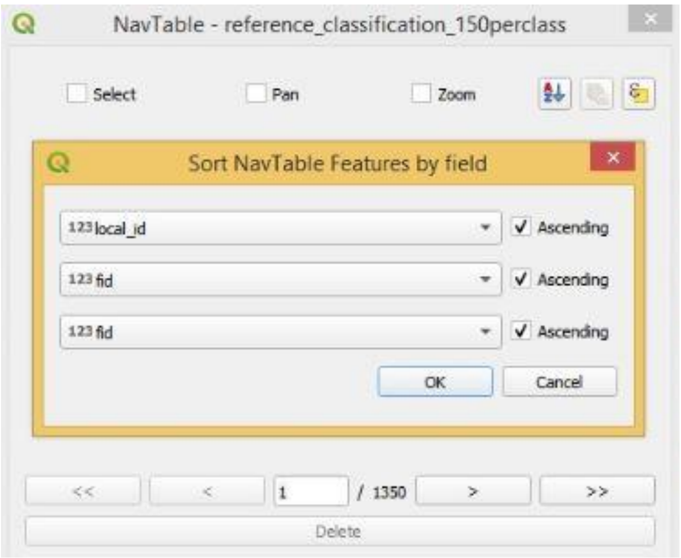
Millimeters

The above fixed settings will allow us to work more efficiently. If we verify the given polygon to be useful it will be symbolized as a green square, in case when we decide otherwise it will disappear because we set rules of symbolisation for the feature with expression „yes” and „NULL” in the „reference” attribute.

To have up to date information about the number of selected polygons for each class we set automatic counting.



Now we will use the **Navtable** plugin for going throughout our attribute table and verify subsequent polygons on the list. Please set sorting features ascending by „**local_id**” attribute.

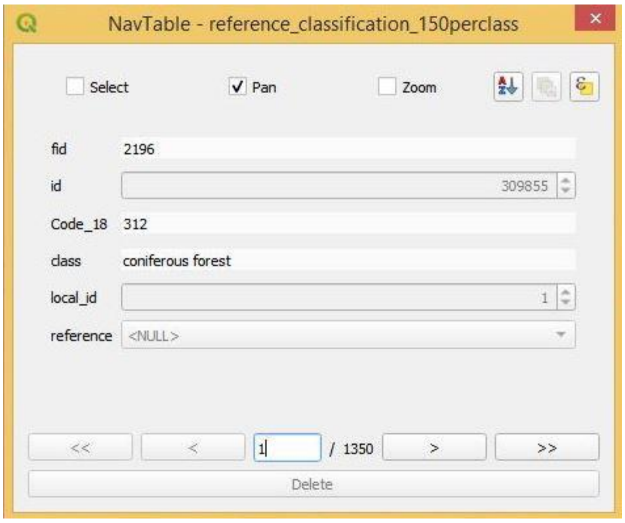

NavTable - reference_classification_150perclass

☐ Select ☐ Pan ☐ Zoom

Sort NavTable Features by field

123 local_id	☑ Ascending
123 fid	☑ Ascending
123 fid	☑ Ascending

In the next window of **Navtable** plugin we will check only „**Pan**” option. When we click a single right arrow view will be moved and centered at the second feature in our table, clicking one more time will move the view to the next one feature and so on. Single left arrow is for moving in the opposite direction.

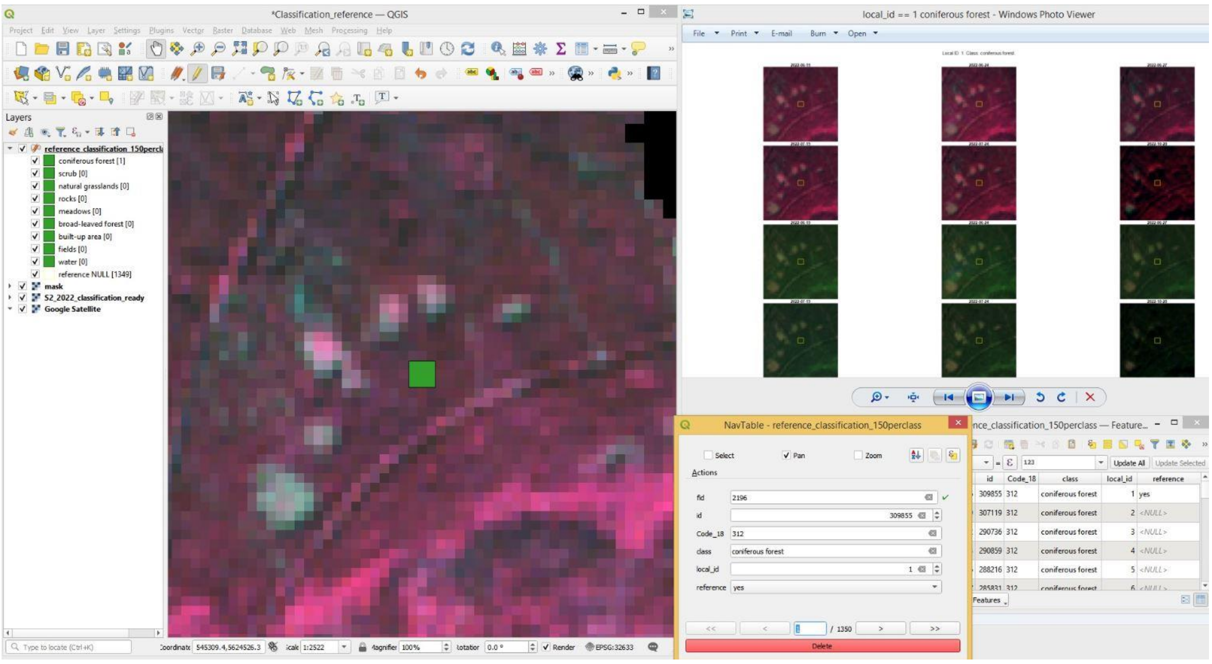


NavTable - reference_classification_150perclass

□ Select ☑ Pan □ Zoom

fid	2196
id	309855
Code_18	312
class	coniferous forest
local_id	1
reference	<NULL>

Now we can start verifying polygons to check if their location is proper (pixels present one land cover class) and it's in accordance with the Corine Land Cover class assigned to it. We will need to open a project in QGIS which we organized up till now, Navtable plugin, attribute table (where we will mark if the given polygon is useful or not) and image chips (where we can verify if polygons are useful in each analyzed date).



*Classification_reference — QGIS

Project Edit View Layer Settings Plugins Vector Raster Database Web Mesh Processing Help
Layers

- ☑ reference_classification_150perclass
  - ☑ 🟩 coniferous forest [1]
  - ☑ 🟩 scrub [0]
  - ☑ 🟩 natural grasslands [0]
  - ☑ 🟩 rocks [0]
  - ☑ 🟩 meadows [0]
  - ☑ 🟩 broad-leaved forest [0]
  - ☑ 🟩 built-up area [0]
  - ☑ 🟩 fields [0]
  - ☑ 🟩 water [0]
  - ☑ reference NULL [1349]
- ☑ mask
- ☑ S2_2022_classification_ready
- ☑ Google Satellite

local_id == 1 coniferous forest - Windows Photo Viewer

File Print E-mail Burn Open

NavTable - reference_classification_150perclass

□ Select ☑ Pan □ Zoom

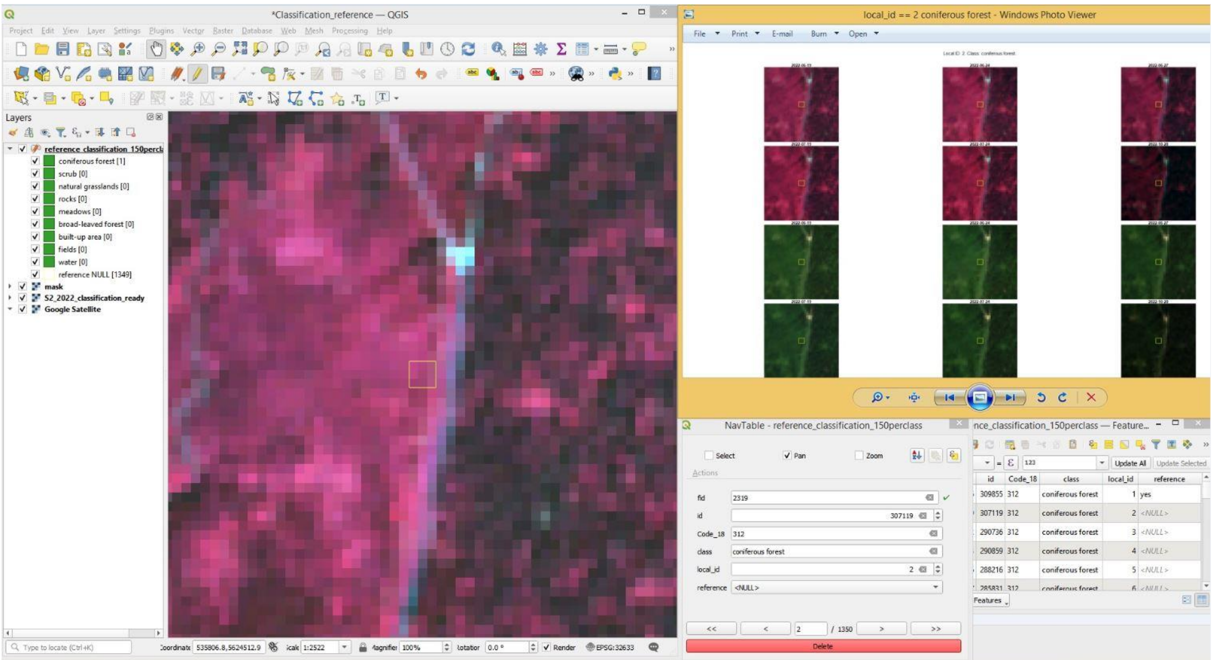
fid	2196
id	309855
Code_18	312
class	coniferous forest
local_id	1
reference	yes

…nce_classification_150perclass — Feature…

id	Code_18	class	local_id	reference
309855	312	coniferous forest	1	yes
307119	312	coniferous forest	2	<NULL>
290736	312	coniferous forest	3	<NULL>
290859	312	coniferous forest	4	<NULL>
288216	312	coniferous forest	5	<NULL>
285831	312	coniferous forest	6	<NULL>

Strategic partnership project funded by the Erasmus+ Program of the European Union (2020-1-CZ01-KA203-078308)

Here is an example of a polygon which location is not proper, but we can shift it by a few pixels and place it in the proper location or we can mark it as not useful and skip it (for that reason we have selected 150 polygons for each class to have a choice). In this example we will choose the first mentioned option.



*Classification_reference — QGIS

local_id == 2 coniferous forest - Windows Photo Viewer

Layers

- ☑ reference_classification_150percla
  - ☑ 🟩 coniferous forest [1]
  - ☑ 🟩 scrub [0]
  - ☑ 🟩 natural grasslands [0]
  - ☑ 🟩 rocks [0]
  - ☑ 🟩 meadows [0]
  - ☑ 🟩 broad-leaved forest [0]
  - ☑ 🟩 built-up area [0]
  - ☑ 🟩 fields [0]
  - ☑ 🟩 water [0]
  - ☑ reference NULL [1349]
- ☑ mask
- ☑ S2_2022_classification_ready
- ☑ Google Satellite

Coordinate 535806.8,5624512.9 Scale 1:2522 Magnifier 100% Rotation 0.0 ° ☑ Render EPSG:32633

NavTable - reference_classification_150perclass

□ Select ☑ Pan □ Zoom

fid	2319
id	307119
Code_18	312
class	coniferous forest
local_id	2
reference	<NULL>

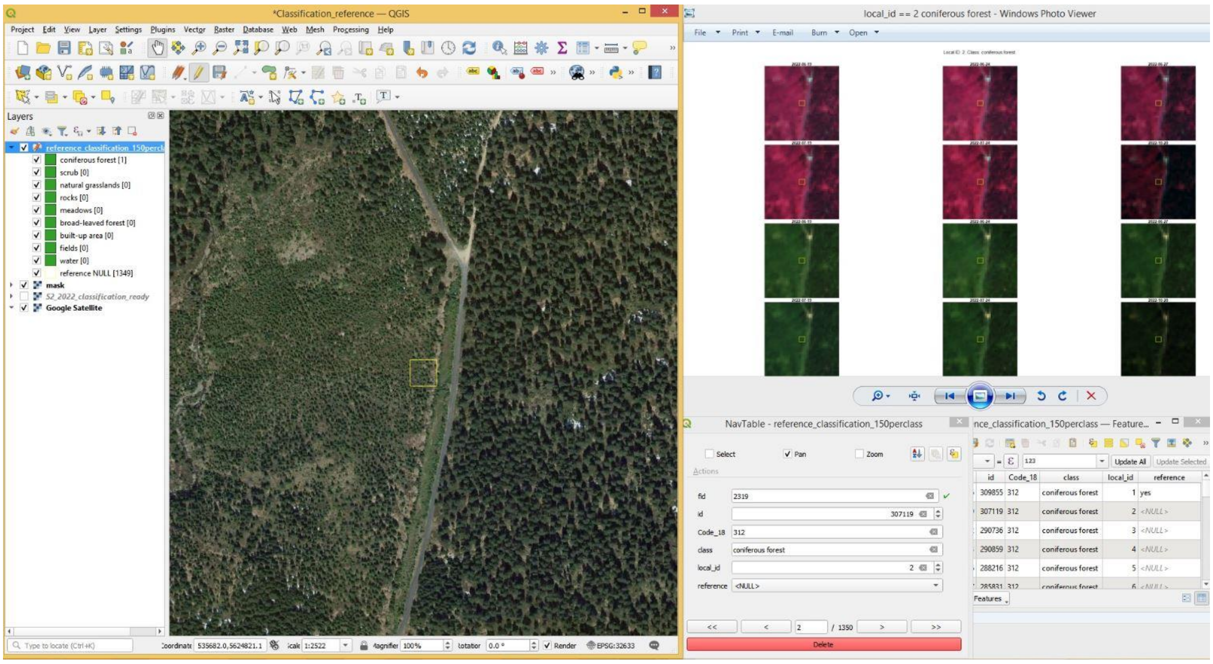
<< < 2 / 1350 > >>

Delete

nce_classification_150perclass — Feature...

id	Code_18	class	local_id	reference
309855	312	coniferous forest	1	yes
307119	312	coniferous forest	2	<NULL>
290736	312	coniferous forest	3	<NULL>
290859	312	coniferous forest	4	<NULL>
288216	312	coniferous forest	5	<NULL>
285831	312	coniferous forest	6	<NULL>

According to the Corine Land Cover our polygon should represent coniferous forest but when we look at CIR composite and Google Satellite basemap we can notice it is located in a place where there are only very young trees.



*Classification_reference — QGIS

local_id == 2 coniferous forest - Windows Photo Viewer

Coordinate 535682.0,5624821.1 Scale 1:2522 Magnifier 100% Rotation 0.0 ° ☑ Render EPSG:32633

Layers

- ☑ reference_classification_150percla
  - ☑ 🟩 coniferous forest [1]
  - ☑ 🟩 scrub [0]
  - ☑ 🟩 natural grasslands [0]
  - ☑ 🟩 rocks [0]
  - ☑ 🟩 meadows [0]
  - ☑ 🟩 broad-leaved forest [0]
  - ☑ 🟩 built-up area [0]
  - ☑ 🟩 fields [0]
  - ☑ 🟩 water [0]
  - ☑ reference NULL [1349]
- ☑ mask
- □ S2_2022_classification_ready
- ☑ Google Satellite

NavTable - reference_classification_150perclass

□ Select ☑ Pan □ Zoom

fid 2319
id 307119
Code_18 312
class coniferous forest
local_id 2
reference <NULL>

<< < 2 / 1350 > >>

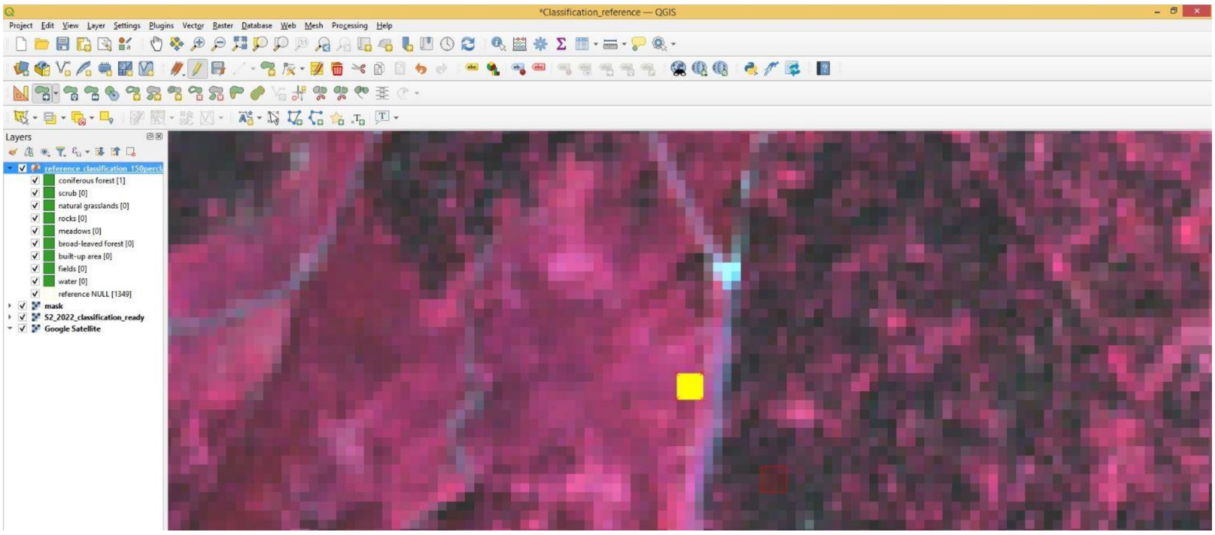
Delete

nce_classification_150perclass — Feature...

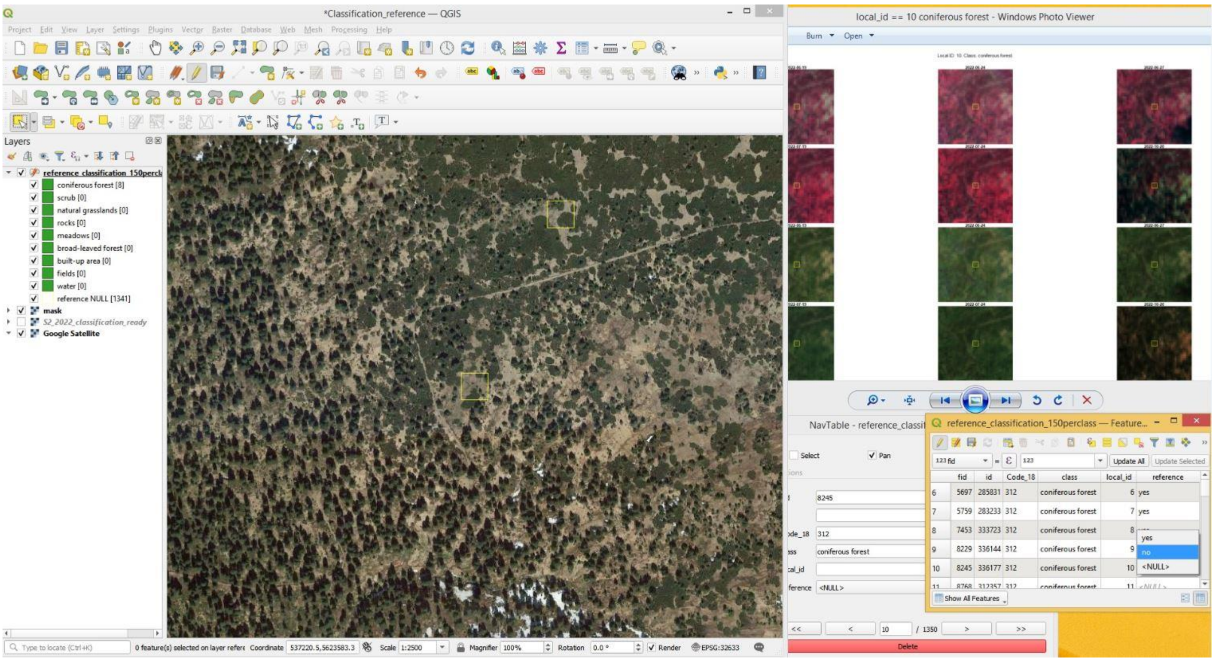
id Code_18 class local_id reference
309855 312 coniferous forest 1 yes
307119 312 coniferous forest 2 <NULL>
290736 312 coniferous forest 3 <NULL>
290859 312 coniferous forest 4 <NULL>
288216 312 coniferous forest 5 <NULL>
285831 312 coniferous forest 6 <NULL>

Strategic partnership project funded by the Erasmus+ Program of the European Union (2020-1-CZ01-KA203-078308)

We will move the polygon in the south-east direction where there is dense coniferous forest.


*Classification_reference — QGIS
Project Edit View Layer Settings Plugins Vector Raster Database Web Mesh Processing Help
Layers
reference_classification_150percl
coniferous forest [1]
scrub [0]
natural grasslands [0]
rocks [0]
meadows [0]
broad-leaved forest [0]
built-up area [0]
fields [0]
water [0]
reference NULL [1349]
mask
S2_2022_classification_ready
Google Satellite

Next example shows a lack of agreement between the class from Corine Land Cover and the actually presented land cover class. In this case we will choose option "no" in attribute "reference" and go further.


*Classification_reference — QGIS
Project Edit View Layer Settings Plugins Vector Raster Database Web Mesh Processing Help
Layers
reference_classification_150percla
coniferous forest [8]
scrub [0]
natural grasslands [0]
rocks [0]
meadows [0]
broad-leaved forest [0]
built-up area [0]
fields [0]
water [0]
reference NULL [1341]
mask
S2_2022_classification_ready
Google Satellite
local_id == 10 coniferous forest - Windows Photo Viewer
Burn Open
NavTable - reference_classif
reference_classification_150perclass — Feature...
Update All
Update Selected
Show All Features
Delete
Scale 1:2500
Magnifier 100%
Rotation 0.0 °
Render
EPSG:32633

NavTable:

☐ Select ☑ Pan

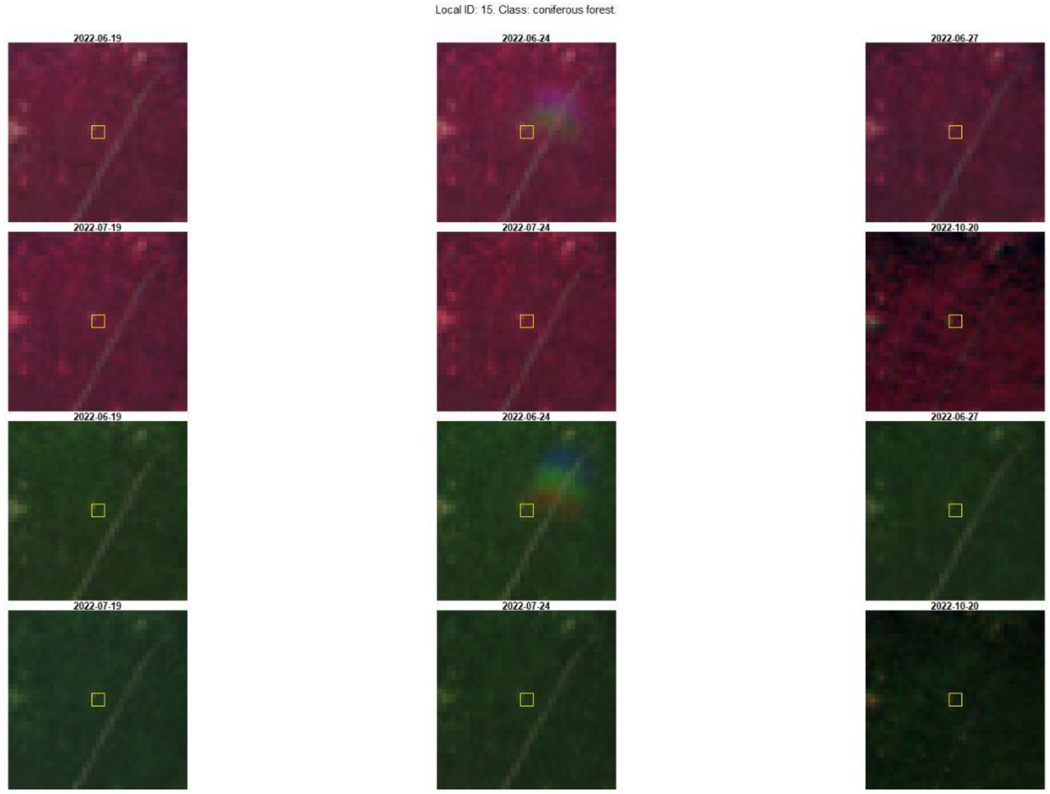
fid	8245
id	
Code_18	312
class	coniferous forest
local_id	
reference	<NULL>

<< < 10 / 1350 > >>

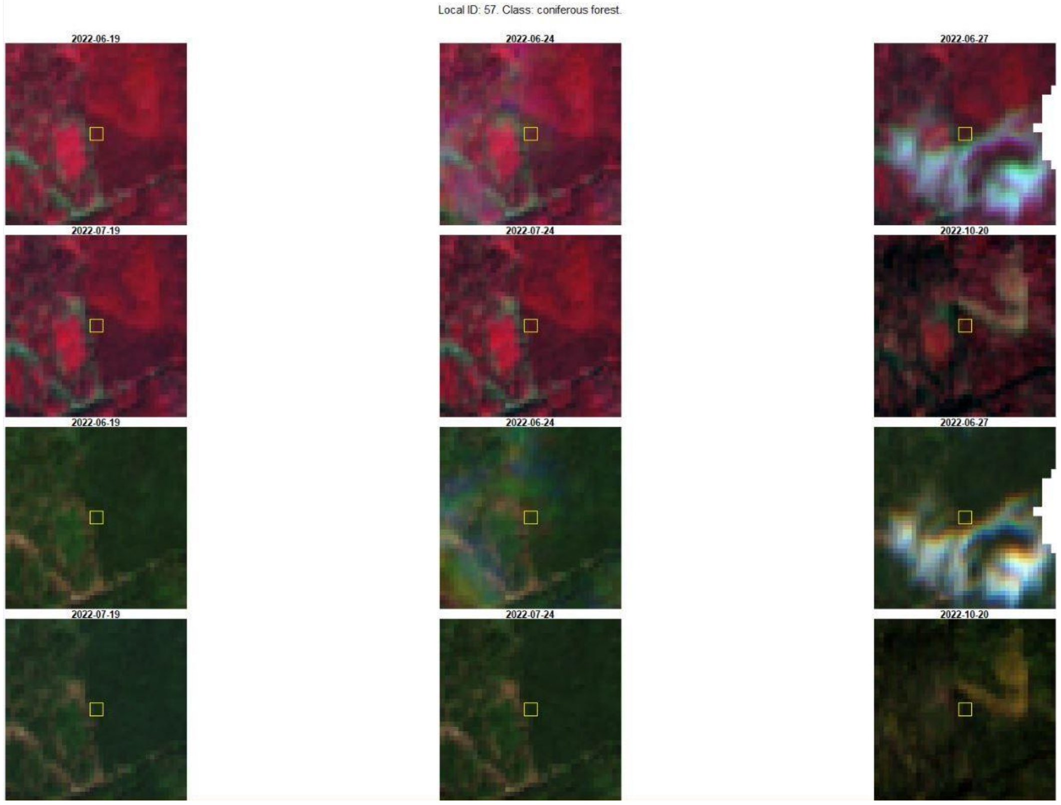
	fid	id	Code_18	class	local_id	reference
6	5697	285831	312	coniferous forest	6	yes
7	5759	283233	312	coniferous forest	7	yes
8	7453	333723	312	coniferous forest	8	yes
9	8229	336144	312	coniferous forest	9	
10	8245	336177	312	coniferous forest	10	
11	8768	312357	312	coniferous forest	11	<NULL>

yes
no
<NULL>

Another reason to reject the polygon would be a visible artifact in the image, in this example visible on image acquired 24.06.2022.



Sometimes the rejection reason may be failure of cloud masking.



If you could not find 50 good samples for each class you can randomly select new polygons to verify and add it to our layer or you can use image interpretation and shift a few rejected before or draw new polygons. For that reason, we have created composites for all acquisition dates. After you obtain the right amount of proper polygons, you should select all good samples and export them as a new layer named “T4_reference_data.shp”.

Proposed manner of reference data preparation allows us to look at all images at once using image chips but like always it is not the only way to realize such a task.

Strategic partnership project funded by the Erasmus+ Program of the European Union (2020-1-CZ01-KA203-078308)