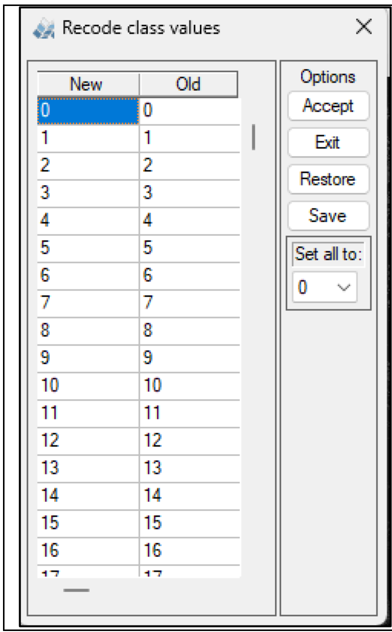
GTB-tools in container: Quick Access
Task: shortcut to essential tools

Recode Classes

(Shortcut to: ***General Tools → Preprocessing → Recode Classes***)

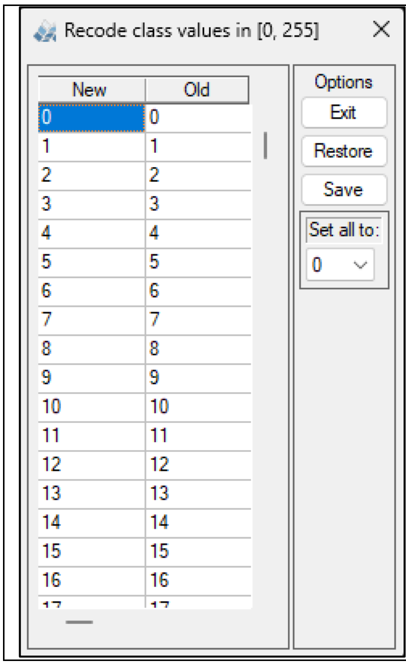


Scope: recode pixel values of the <u>currently loaded map</u>.
Displays a table showing new (left) and current unique class values (right).

- This option is only available for images of data type Byte. Please note that any new value in the cell of the left column will only be taken over after **pressing the Enter key** or after switching to a different cell of the left column.
- Use the *Set all to* dropdown menu to assign all classes to a single value.
- Click the *Save* button to save a new recode table to a file **GTBrecode_*.sav**. Keep the prefix ***GTBrecode_*** and the file extension ***.sav*** and only modify the placeholder *. Any previously saved recoding table **GTBrecode_*.sav** can be restored via the *Restore* button. Here, only those entries that match the current class values will be restored.
- Finally, click the *Accept* button to apply the recoding table to the <u>currently loaded map</u>.

Setup Batch Recode Table

(Shortcut to: ***General Tools → Preprocessing → Setup Batch Recode Table***)



Scope: define/amend a recode table for <u>batch-recoding multiple maps</u>.
Displays the full Byte range of 256 values in [0, 255] with the new values (left) and all potential old values (right).

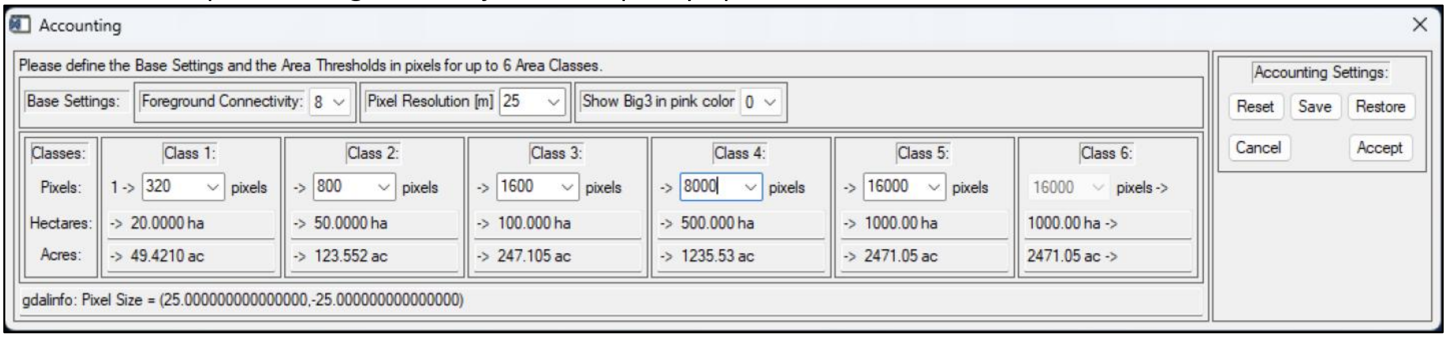
- This option is only available for images of data type Byte. Please note that any new value in the cell of the left column will only be taken over after **pressing the Enter key** or after switching to a different cell of the left column.
- Use the *Set all to* dropdown menu to assign all classes to a single value.
- Click the *Save* button to save a new recode table to a file **GTBbatchrecode_*.sav**. Keep the prefix ***GTBbatchrecode_*** and the file extension ***.sav*** and only modify the placeholder *. Any previously saved recoding table **GTBbatchrecode_*.sav** can be restored via the *Restore* button.
- Once done, use ***File → Batch Process → Recode Classes*** and restore your previously saved recoding table **GTBbatchrecode_*.sav** for <u>batch-recoding multiple maps.</u>

Accounting (← click for product sheet)

(Shortcut to: ***Image Analysis → Objects → Accounting***)

Question: what is the size class distribution of the foreground objects (2 byte)?

How: load a map with foreground objects and specify up to six size class thresholds:

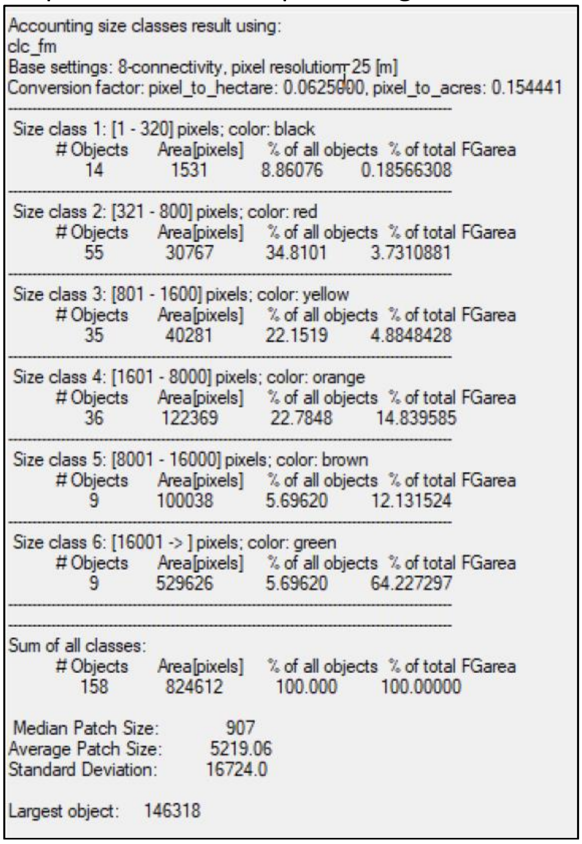


The chart above shows six user-selected thresholds in number of pixels, which are automatically converted into patch size classes up to 20, 50, 100, 500, 1000, >1000 hectare.

Result:

Statistics: number, area, and proportion of foreground objects in each size class; total number/area of foreground patches, average patch size and area of largest foreground object.

Map: color-coded map showing location of foreground objects in each size class.



Accounting size classes result using:
clc_fm
Base settings: 8-connectivity, pixel resolution: 25 [m]
Conversion factor: pixel_to_hectare: 0.0625000, pixel_to_acres: 0.154441

Size class 1: [1 - 320] pixels; color: black

# Objects	Area[pixels]	% of all objects	% of total FGarea
14	1531	8.86076	0.18566308

Size class 2: [321 - 800] pixels; color: red

# Objects	Area[pixels]	% of all objects	% of total FGarea
55	30767	34.8101	3.7310881

Size class 3: [801 - 1600] pixels; color: yellow

# Objects	Area[pixels]	% of all objects	% of total FGarea
35	40281	22.1519	4.8848428

Size class 4: [1601 - 8000] pixels; color: orange

# Objects	Area[pixels]	% of all objects	% of total FGarea
36	122369	22.7848	14.839585

Size class 5: [8001 - 16000] pixels; color: brown

# Objects	Area[pixels]	% of all objects	% of total FGarea
9	100038	5.69620	12.131524

Size class 6: [16001 ->] pixels; color: green

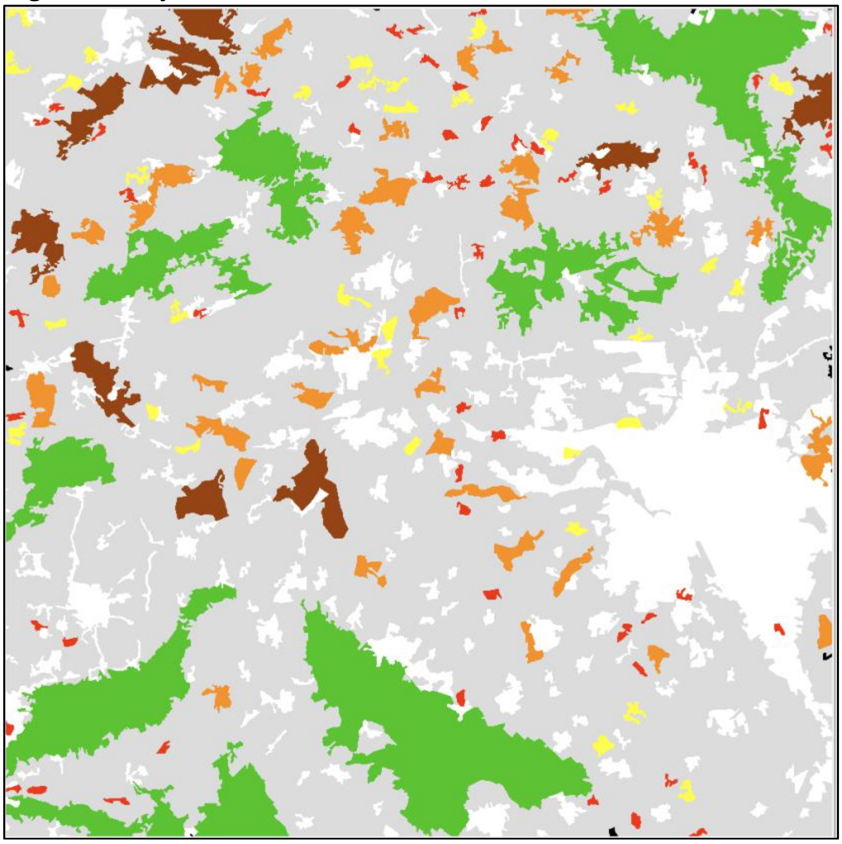
# Objects	Area[pixels]	% of all objects	% of total FGarea
9	529626	5.69620	64.227297

Sum of all classes:

# Objects	Area[pixels]	% of all objects	% of total FGarea
158	824612	100.000	100.00000

Median Patch Size: 907
Average Patch Size: 5219.06
Standard Deviation: 16724.0

Largest object: 146318

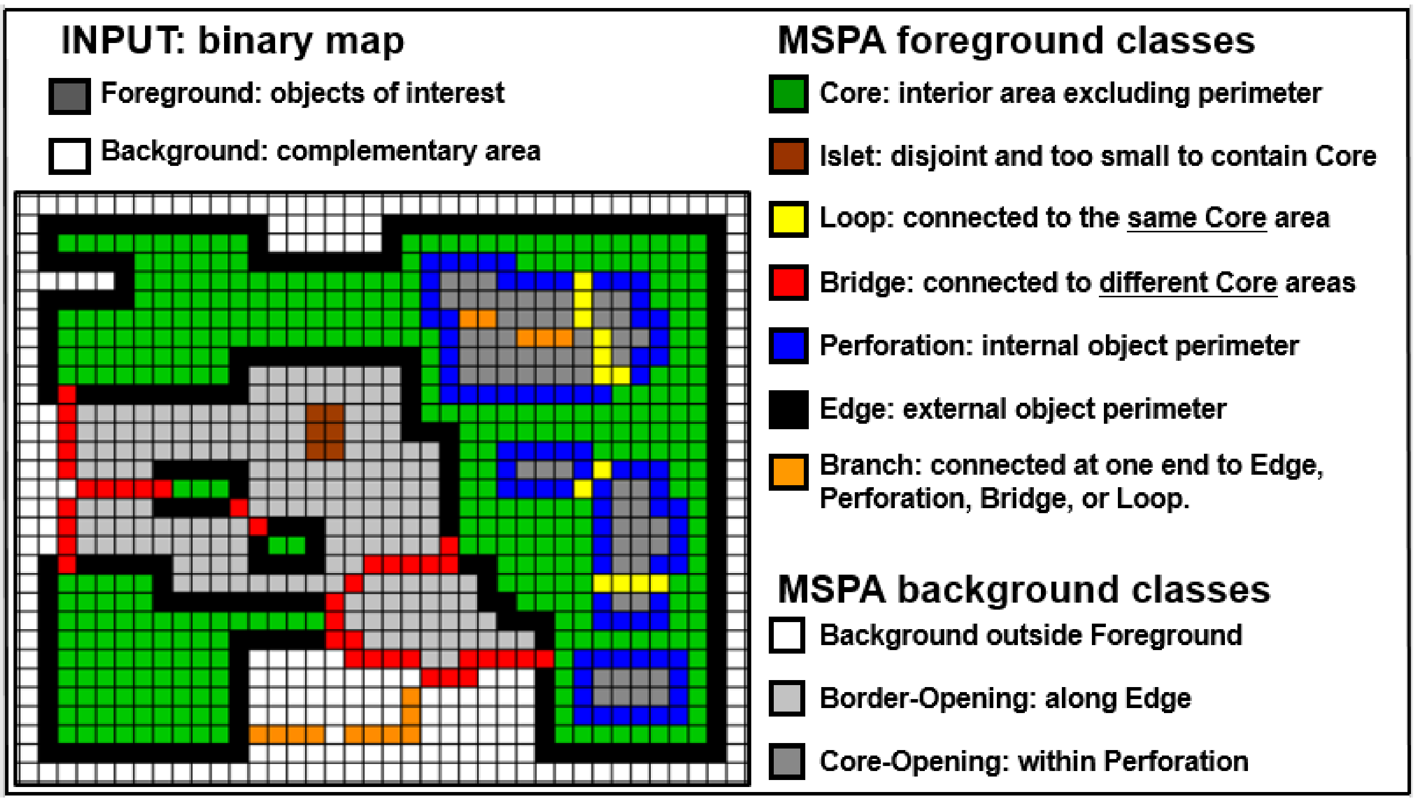


The chart above shows the statistics and the map of the six size classes color-coded in (black-103, red-33, yellow-65, orange-1, brown-9, green-17). Moving the mouse cursor in the GTB viewport will show the ID and area in pixels for each foreground object.

Pattern (← click for product sheet)
(Shortcut to: ***Image Analysis → Pattern → Morphological → MSPA***)

Question: what are the morphological features of the foreground objects (2 byte)?

We use a customized sequence of mathematical morphological operators to describe the geometry and connectivity of the foreground image objects. Based on geometric concepts only, this method can be applied at any scale and to any type of digital maps in any application field.



The chart above shows the Morphological Spatial Pattern Analysis (MSPA) providing up to 10 visual classes and up to 23 numerical feature classes. All features and processing options are summarized in the dedicated MSPA Guide or can be found on the MSPA website.
Note: The alternative Simplified Pattern Analysis (SPA) with up to 6 feature classes may be sufficient and is available from the menu entry: ***Image Analysis → Pattern → Morphological → SPA2-6***.

Fragmentation (← click for product sheet)

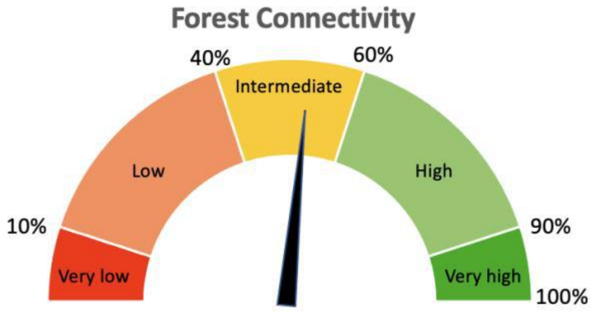
(Shortcut to: ***Image Analysis → Fragmentation → Fixed Observation Scale (FOS)***)

Question: what is the degree of fragmentation/connectivity of the foreground pixels (2 byte)?

Connectivity, or its complement fragmentation, is scale dependent. The size of the scale of interest is defined via the size of the local neighborhood that is analyzed when measuring the degree of connectivity at pixel level. Think of it as opening an umbrella over a foreground pixel, do the assessment for the area covered under the umbrella (local neighborhood), and assign the result to the center pixel of the umbrella in the output map. The assessment itself measures the *Foreground Area Density* (FAD), which is the number of foreground pixels with respect to the total number of pixels in the local neighborhood. This focal or moving window analysis is repeated over every foreground pixel, or the umbrella is moved and opened over each foreground pixel.
The result of the moving window analysis is a map with the same foreground coverage but showing the degree of connectivity in percent within the neighborhood (under the umbrella) over each foreground pixel. The resulting map of forest connectivity at pixel level can be reported in various ways at:

a) pixel level: report each foreground pixel connectivity value, or
b) pixel level: same as a) but color-coded into several categories of connectivity/fragmentation, or
c) patch level: build the average of the foreground pixel connectivity values for each patch, or
d) reporting unit level: a single value, the average over all foreground pixel connectivity values.

Example for reporting forest connectivity/fragmentation using the default 5-class reporting scheme:



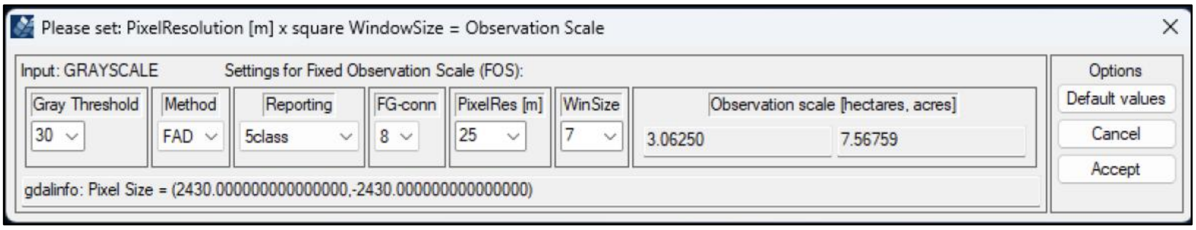
Foreground cover	Color	FAD/FAC	Connectivity	Fragmentation
1-Rare		0% ≤ x < 10%	Very low	Very high
2-Patchy		10% ≤ x < 40%	Low	High
3-Transitional		40% ≤ x < 60%	Intermediate	Intermediate
4-Dominant		60% ≤ x < 90%	High	Low
5-Interior		90% ≤ x ≤ 100%	Very high	Very low

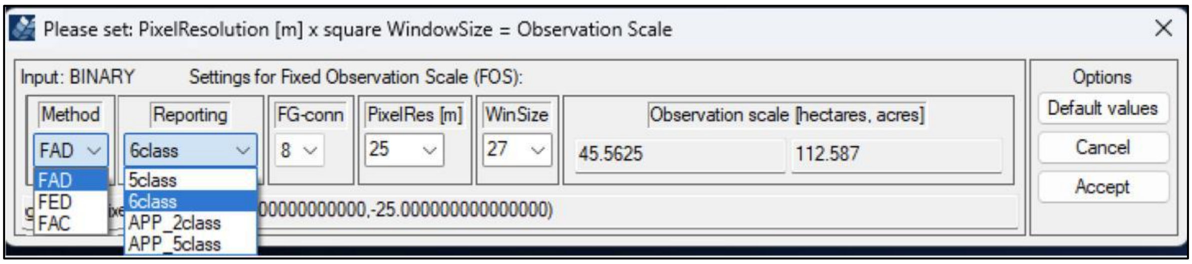
For example, in class 4 the foreground cover within the neighborhood is *dominant*, which is equivalent to FAD or FAC being within the range [60, 90]%, the degree in connectivity is *high*, and the degree in fragmentation is *low*. The 6-class reporting scheme is similar but with a separate class 6, named *Interior*, where FAD/FAC = 100%.

The 2-class reporting scheme uses a threshold of 40% to divide the full range of connectivity into the two classes *Separate* and *Continuous* foreground cover:

Foreground cover	Color	FAD/FAC	Connected	Fragmented
1-Separated		0% ≤ x < 40%	No	Yes
2-Continuous		40% ≤ x ≤ 100%	Yes	No

How: load a grayscale map (top) or a binary map (bottom) with foreground objects (2 byte) and specify:





- Grayscale Threshold (only available when using grayscale input maps):
  Grayscale input data has foreground values within the range [0, 100], for example a tree cover density map. Here, background areas are defined where pixel values are below a user-selected foreground grayscale threshold value *grayt*. For example, if grayt=30 then background pixels are in [0, 29] and foreground pixels are in [30, 100]. When using grayscale input maps, missing data must be encoded with 255 byte.

- FOS Method:
  - FAD (Foreground Area *Density* – proportion of foreground pixels); FED (Foreground Edge *Density* – proportion of foreground edges/all edges); or FAC (Foreground Area *Clustering* – proportion of common adjacencies between neighboring foreground pixels)
  - Reporting: the fragmentation range [0, 100]% is color-coded and reported at pixel-level with 5 or 6 classes, or at patch-level (APP - Average-Per-Patch) with 2 or 5 classes
- FG-conn: 8 – all directions (default) or 4 – connectivity in horizontal/vertical directions only
- PixelRes: spatial pixel resolution in meters
- WinSize: side length (number of pixels) of local neighborhood square window or umbrella.
- Observation scale: (PixelRes * WinSize)^2 = the area of the local neighborhood to be considered when assessing fragmentation/connectivity.

Result:
Statistics: proportion of foreground pixels in each fragmentation class; average degree in fragmentation.
Map product: showing fragmentation at pixel-level, color-coded into the selected number of classes.

Landscape Mosaic (← click for product sheet)

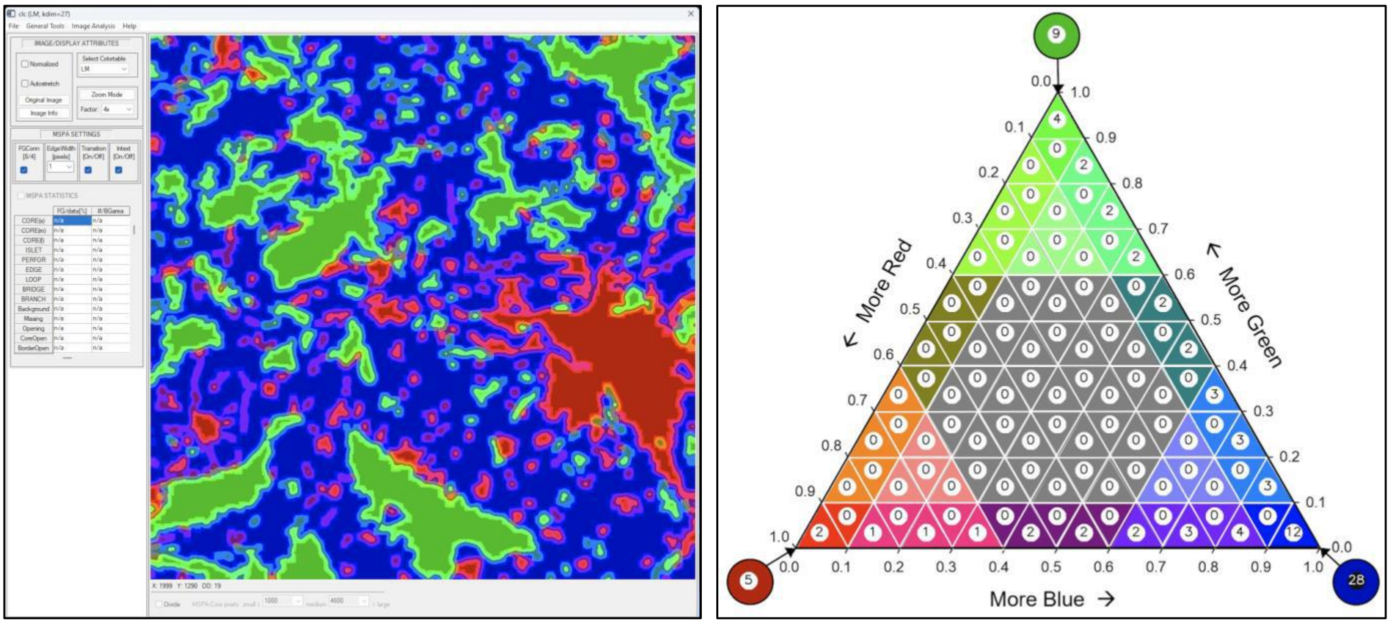
(Shortcut to: ***Image Analysis → Pattern → Moving Window → Counting Pixels→ Landscape Mosaic***)

Question: what is dominant landcover type in the local neighborhood?

The Landscape Mosaic (LM) is a tri-polar classification of a location accounting for the relative contributions of the three landcover types of *Agriculture*, *Natural*, and *Developed* within the local neighborhood surrounding that location. The classification model is designed to identify anthropogenic activity (landcover classes falling in the categories *Agriculture* and *Developed*) in relation to *natural* landcover. The analysis scale is driven by the size of the moving window (local neighborhood).

How: load a LM-compatible input map (a Byte array with no more than 3 target classes with the assignment AND (1-Agriculture, 2-Natural, 3-Developed, plus an optional class for missing values: 0 Byte) and run the LM analysis.

Result:
Statistics: heatmap summary or proportion of occurrence frequency in each LM class.
Map: color-coded map showing Landscape Mosaic classes for each pixel at the user-selected observation scale.



The chart above shows an example of the Landscape Mosaic processing. The left panel shows the mosaic classes: each pixel or cell on the map has a certain contribution of Agriculture, Natural, Developed landcover within its local neighborhood of 27 x 27 pixels. The right panel shows the occurrence frequency distribution of all pixels: for example, the green circle at the top of the triangle implies that 9% of all pixels in the Landscape Mosaic map are fully natural, meaning their local neighborhood is composed of natural landcover only. Many pixels in the map have a certain contribution of Agriculture, Natural, Developed landcover in their local neighborhood, which is represented by a respective sub triangle within the Landscape Mosaic triangle. Further details on processing options and application examples can be found in the **Landscape Mosaic product sheet**.