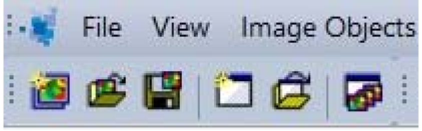
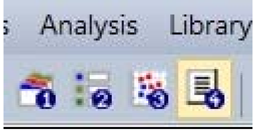
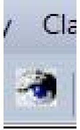
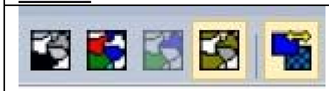
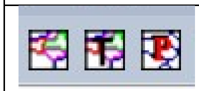
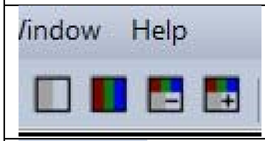
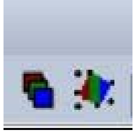
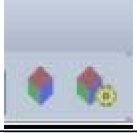
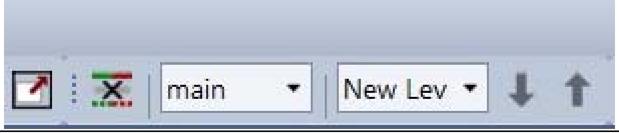
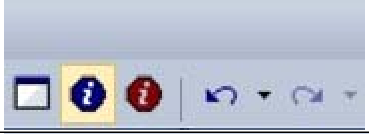
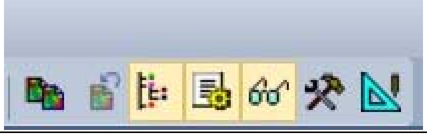
NOTES from eCognition Workshop

Anatomy of the Main Menu Bar

Buttons	Function
	Buttons for creating/opening projects and workspaces. N.B. in the eCog world, projects are nested within workspaces...
	Selection of analysis window. Stage #4, for example, loads up a set of tiled windows for “Process Properties”, “Process Tree”, “Class Hiearchy”, etc. 1 = Load and manage data, 2 = configure analysis, 3 = Review Results, 4 = Develop Rulesets
	View settings
	Controls the display of (1) the original raw imagery (there’s another button you can use to toggle through individual bands), (2) classified objects, (3) selected objects (if a ML-based classification procedure is to be used)
	Control the display of object outlines; T = sets transparency on/off; P = turns off/on polygons
	
	Control the blending/mixing of raster bands (button on left), or presentation of vector data (e.g., polygon colour, =funky spaceship looking icon)
	
	

	
	These allow you to open closed processing pane windows, e.g., the script with a gear symbol refers to the Rule set window.

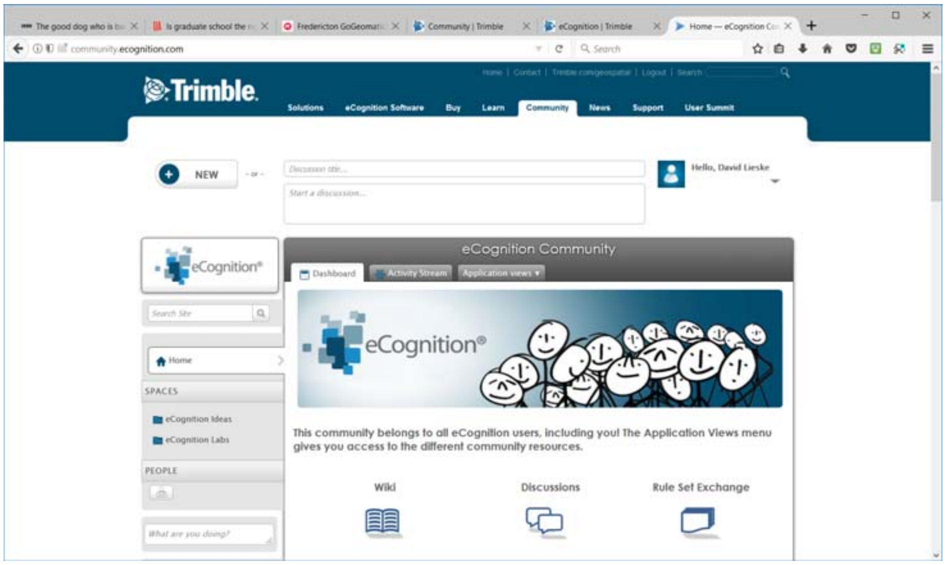
Key Resources:

- The **User's guide** acts as an operating manual; the **reference book** as an encyclopedia.
- The eCognition Community is a good reference as well; users can access free of charge.
- It's an integrated environment for discussions, uploads, downloads, etc.
- You can also provide product feedback, and there's social networking facilities.
- Not a support portal, though.
- If you need help, please identify yourself and provide a detailed description of the problem, along with screenshots.
- http://letters-sal.blogspot.de (Jarlath O'Neil-Dunne)

- Need maintenance? maintenance@ecognition.com, or Support@eCognition.com

- You can check out more examples on the eCognition Community link:

  http://www.ecognition.com/community



This community belongs to all eCognition users, including you! The Application Views menu gives you access to the different community resources.

OCT.17, 2017

- Instructors: Keith Peterson (Product Manager; originally from Vermont, USA) and Matthias Staengel; LULC – land use land cover = Matthias Staengel (Masters in Applied Geoinformatics): keith_peterson@trimble.com, matthias_staengel@trimble.com
- Fundamentals training – complete image analysis workflow; introduction to tools and features.
- Analysis strategies training- more production-oriented training.
- **Inpho** – Trimble program for pre-processing of imagery.
- eCog Developer = object-oriented image analysis; constructs rule sets; executes locally.
- eCog Architect = provides an easy-to-use front end for non-eCog users to interact with.
- eCog Server = supports batch processing.

- eCog can combine data from multiple sources, e.g., vector data, DEM point cloud, rasters to produce superior results.
- Key step is the Rule Set Development.
- eCognition Server allows for batch processing, dynamic load balancing, service-oriented architecture, highly scalable. Can do parallel processing, but need two licenses (one per processor).
- Application areas are typically forest classification, urban, and ag, but also offshore (e.g., sea floor mapping). Defense and Security (e.g., fire detection) is another big one.
- Paper: “German engineering: Imagery tools automate vegetation mapping in the Rhineland”.

- Paper: “A versatile, production-oriented approach to high-resolution tree-canopy mapping in urban and suburban landscapes...”
- Agro-forestry: Tree counting in palm oil plantations in SE Asia
- Application: Development of 2010 national land cover database for the Nepal.
- In Austria: achieved 95% accuracy for automatic vegetation and building extraction with high-resolution imagery.
- Paper: “Automatic three-dimensional features extraction: the case study of L’Aquila for collapse identification after April 6, 2009 earthquake” (recognition success: 93%, compared to pixel-based approaches of 64%).

- V9 was released around 2014.
- There’s some more advanced data support (e.g., MrSID, LAZ files) in v 9.2.
- Big emphasis on machine learning: use of neural networks, etc.
- Google TensorFlow library
- Point cloud classification – rasterizing point clouds, assigning classes to point clouds, export, etc.
- Point Cloud Input (*.LAS) – use case for generating DTMs. The point cloud gets rasterized (e.g., with maximum return), and then is subject to classification: ground, buildings, trees, power poles (?)
- Use case: change detection (I)

- What is a **Convolutional Neural Network (CNN)** “deep learning”? Filters are applied at teach traning image at different resolutions and output of each convolved image is used as the input to the next layer to identify features. Can lead to some very low error rates compared to template matching (6.8% vs. 22.2%).

- License handling based on files is replaced with **Trusted Storage**. You borrow and activate from the FlexNet cloud (?)
- Big development area has been to combine vector and raster data formats.

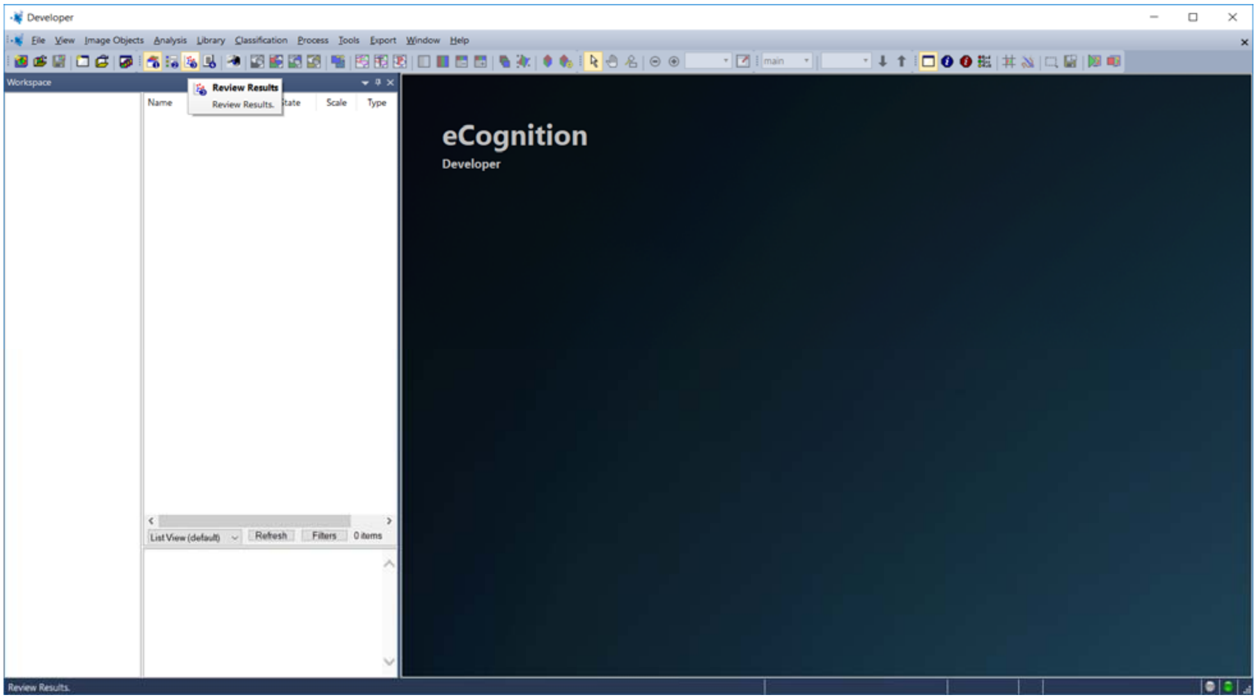
- **OBIA = Object-based Image Analysis**: colour, tone, texture, etc. A pixel is the information carrier.Pixels do not contain enough information to describe what they see, e.g., water in a river, or a lake? O’Niel-Dunne 2009 – objects more meaningful when shape and context are involved.
- For instance, buildings that border on shadow objects can be inferred to have height – that can be used to distinguish a building roof from a paved road surface, for instance.
- Image analysis workflow: based on **Rule sets**. Contains all the processes or algorithms you need to extract and export objects of interest. Segmentation, classification steps, reshaping and refinement are all sequences to finish your classification.
- **CNL** = Cognition Network Language – the language for translating human recognition process.

- **Rule sets** can be stored in ***.dcp** file format, and can have parent-child nested relationships.

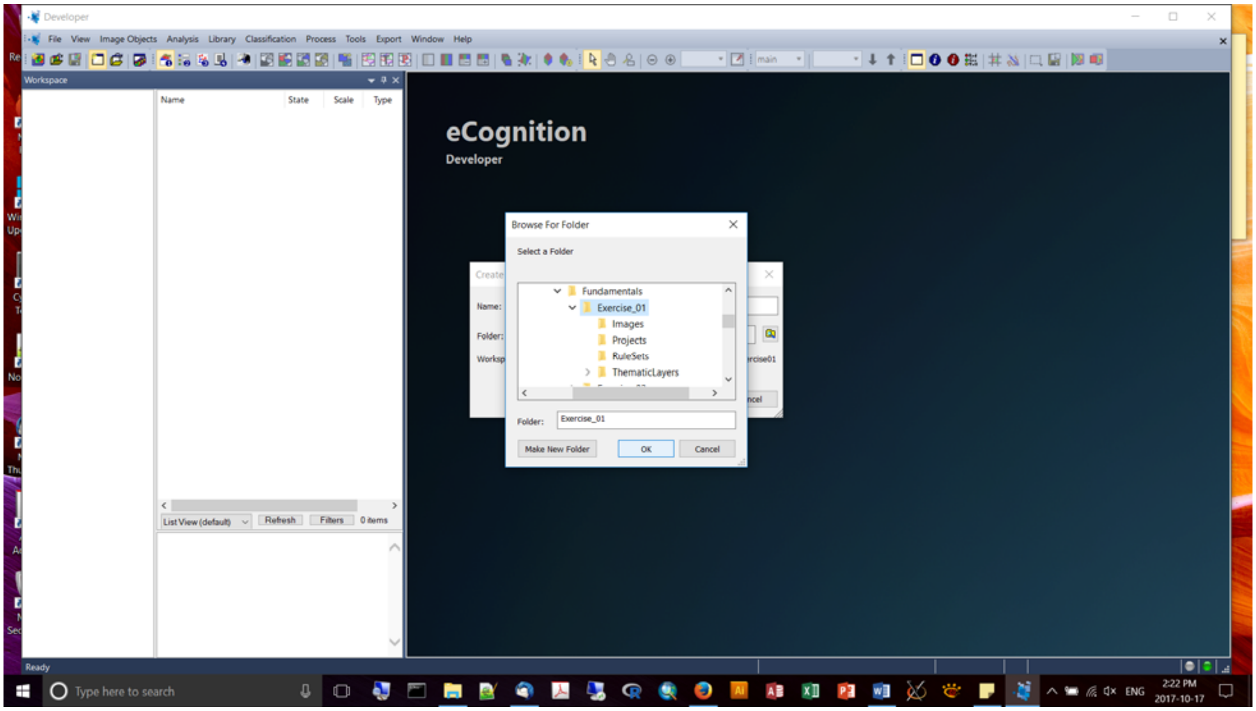
- Data Handling: can combine data from various sensors and platforms within eCognition Projects. Projects saved as ***.dpr** files.
- A workspace acts as a container for projects, and are stored as ***.dpj** files. You need this for Server.

Exercise #1

- 4-band Imagery (RGB & NIR); DSM, DTM, shapefile, FileGDB.
- <u>Prior to working with eCog data, it is important that they share the same coordinate system and overlay correctly. You can't project/re-project data in eCog</u>. Atmospheric correction? Mosaicing? Geo-rectification? The viewing tools are important for inspecting your layers, and for making sure everything is logically named. If you haven't set the names properly the processing downstream may not recognize your layers.

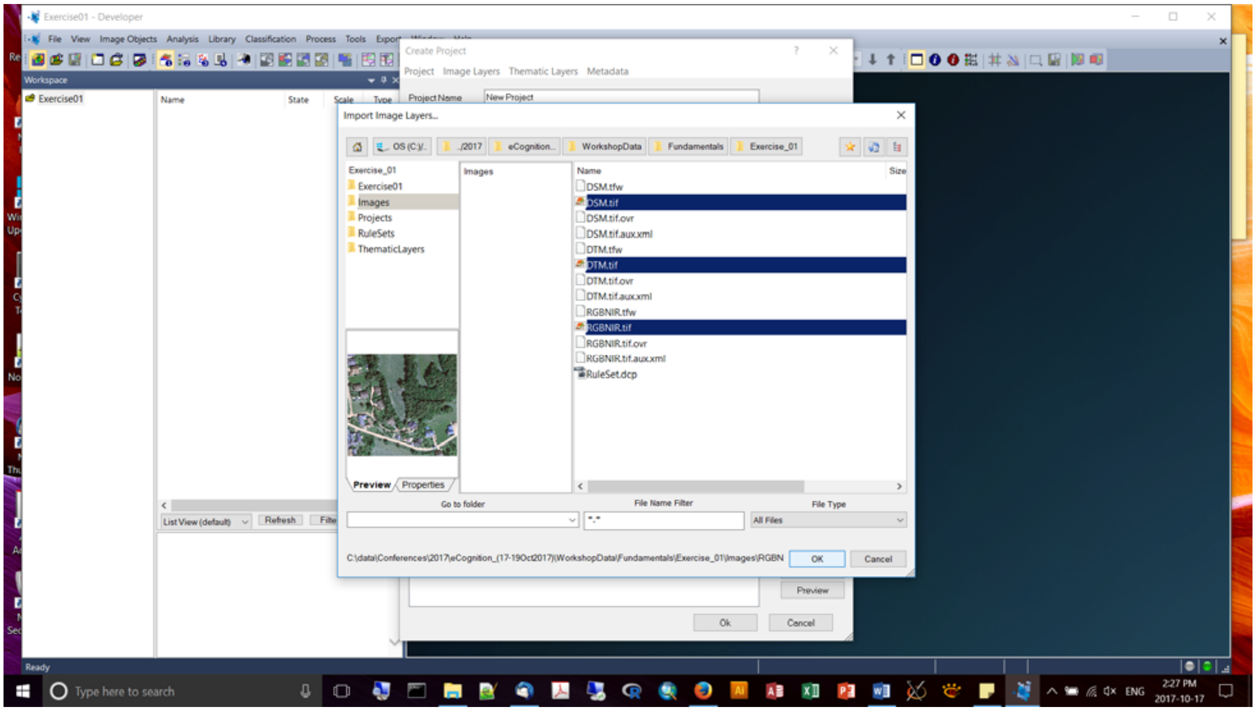


- The "View Settings" buttons allow you to launch a few tasks (number 1-4).
- Start by creating a workspace:



Folder: Exercise_01

- Then create a new project and load the data (as separate .tif files):



Project Name: New Project

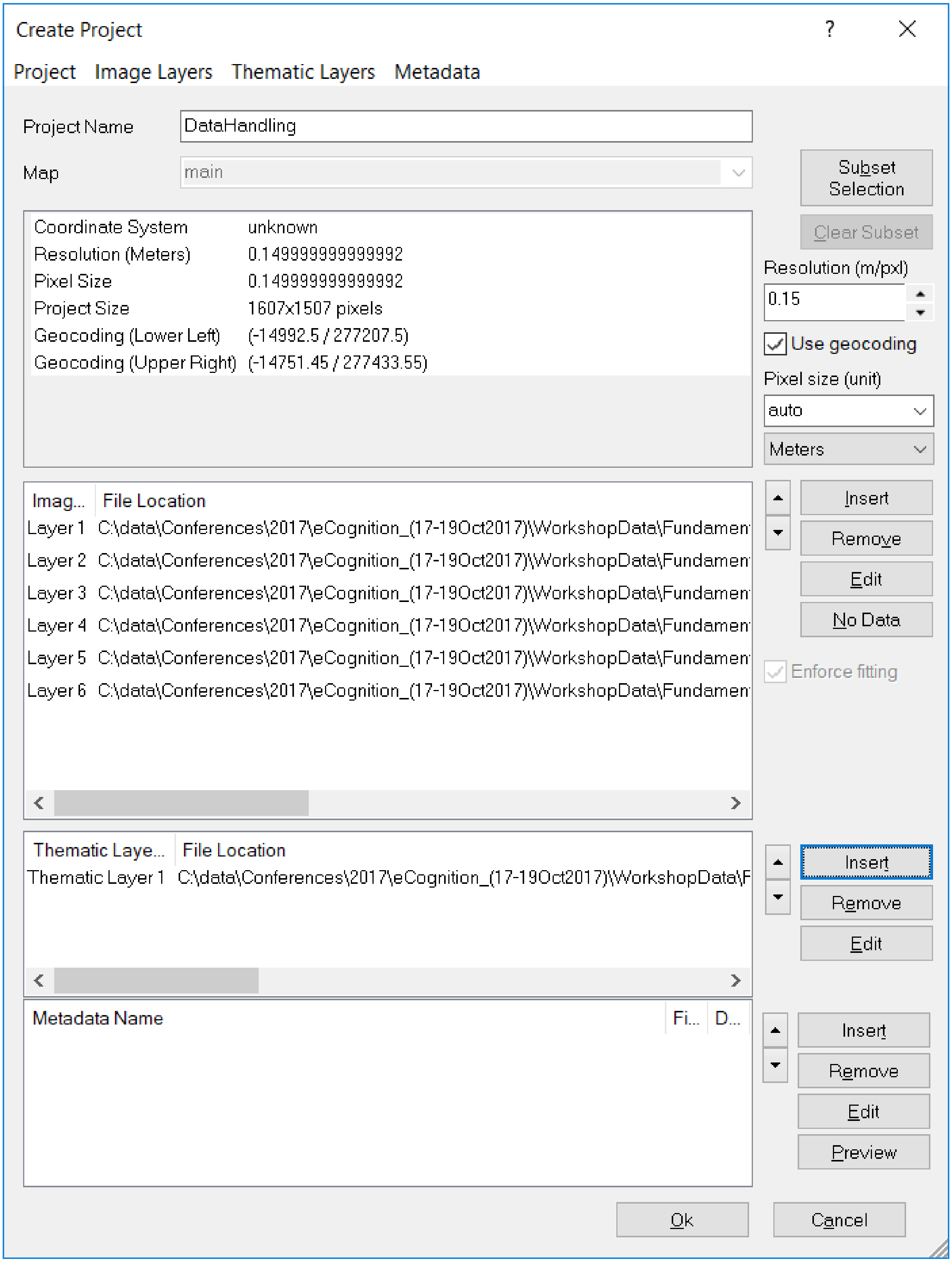
DSM.tfw
DSM.tif
DSM.tif.ovr
DSM.tif.aux.xml
DTM.tfw
DTM.tif
DTM.tif.ovr
DTM.tif.aux.xml
RGBNIR.tfw
RGBNIR.tif
RGBNIR.tif.ovr
RGBNIR.tif.aux.xml
RuleSet.dcp

File Name Filter: *.*

File Type: All Files

C:\data\Conferences\2017\eCognition_(17-19Oct2017)\WorkshopData\Fundamentals\Exercise_01\Images\RGBN

- Then load the shapefile “manholes.shp” (make sure that the show file type is “all’):



Project Name: DataHandling

Map: main

Coordinate System	unknown
Resolution (Meters)	0.1499999999999992
Pixel Size	0.1499999999999992
Project Size	1607x1507 pixels
Geocoding (Lower Left)	(-14992.5 / 277207.5)
Geocoding (Upper Right)	(-14751.45 / 277433.55)

Resolution (m/pxl): 0.15

☑ Use geocoding

Pixel size (unit): auto

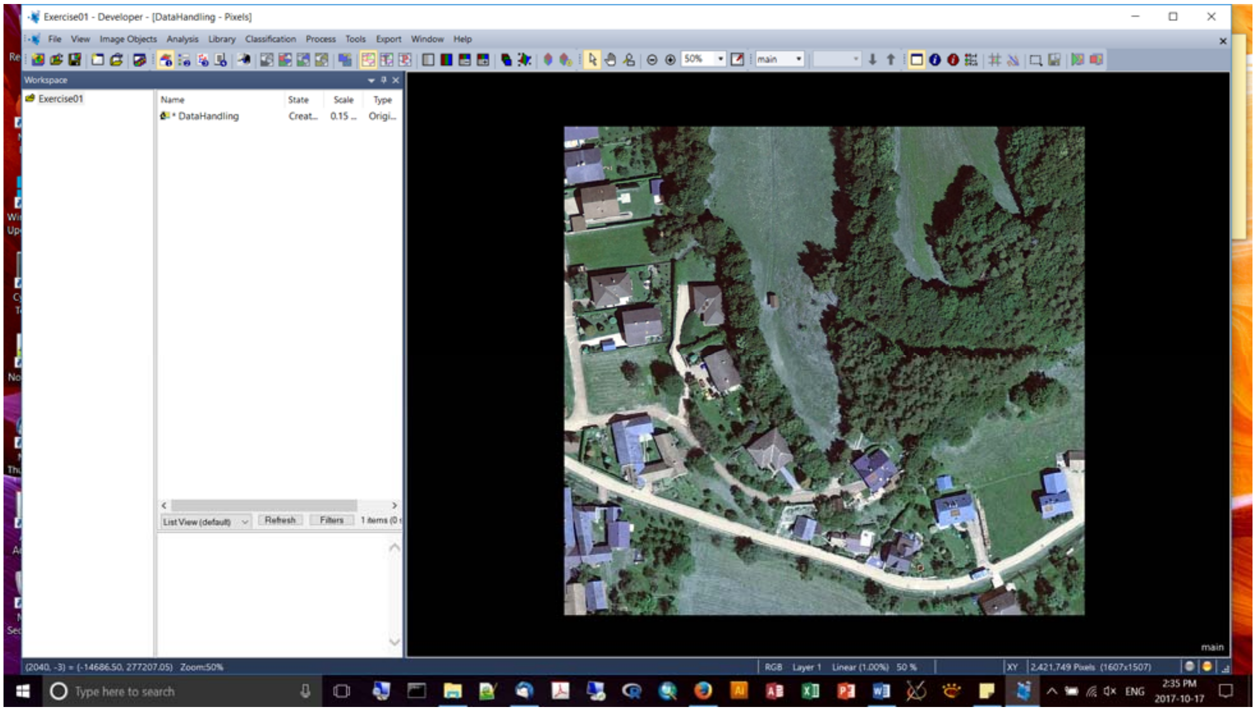
Meters

Imag...	File Location
Layer 1	C:\data\Conferences\2017\eCognition_(17-19Oct2017)\WorkshopData\Fundamen
Layer 2	C:\data\Conferences\2017\eCognition_(17-19Oct2017)\WorkshopData\Fundamen
Layer 3	C:\data\Conferences\2017\eCognition_(17-19Oct2017)\WorkshopData\Fundamen
Layer 4	C:\data\Conferences\2017\eCognition_(17-19Oct2017)\WorkshopData\Fundamen
Layer 5	C:\data\Conferences\2017\eCognition_(17-19Oct2017)\WorkshopData\Fundamen
Layer 6	C:\data\Conferences\2017\eCognition_(17-19Oct2017)\WorkshopData\Fundamen

☑ Enforce fitting

Thematic Laye...	File Location
Thematic Layer 1	C:\data\Conferences\2017\eCognition_(17-19Oct2017)\WorkshopData\F

Metadata Name	Fi...	D...

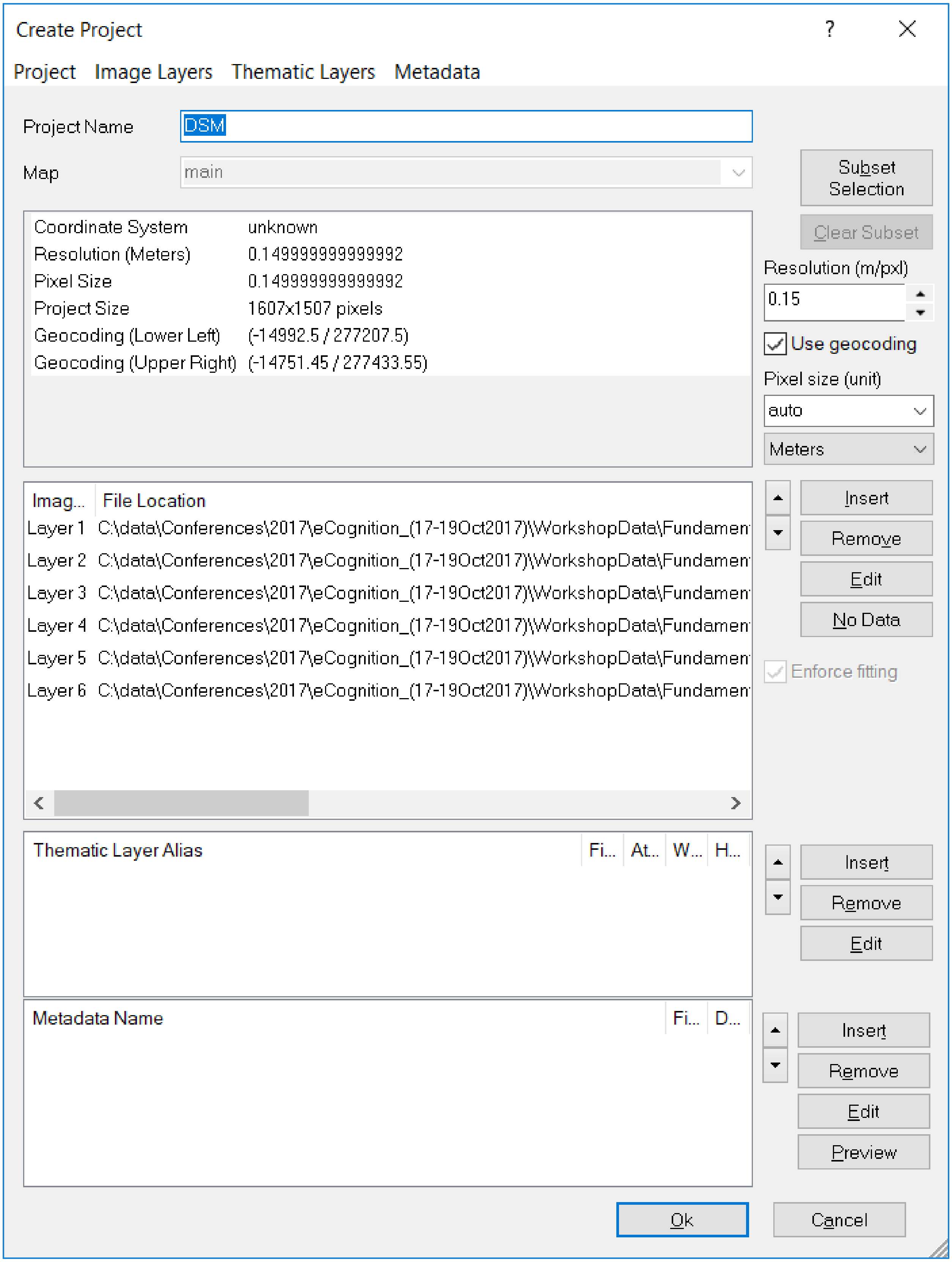


Name	State	Scale	Type
* DataHandling	Creat...	0.15 ...	Origi...

(2040, -3) = (-14686.50, 277207.05) Zoom:50%

RGB Layer 1 Linear (1.00%) 50 %

XY 2,421,749 Pixels (1607x1507)



Create Project

Project Name: DSM

Map: main

Coordinate System	unknown
Resolution (Meters)	0.1499999999999992
Pixel Size	0.1499999999999992
Project Size	1607x1507 pixels
Geocoding (Lower Left)	(-14992.5 / 277207.5)
Geocoding (Upper Right)	(-14751.45 / 277433.55)

Resolution (m/pxl): 0.15

☑ Use geocoding

Pixel size (unit): auto

Meters

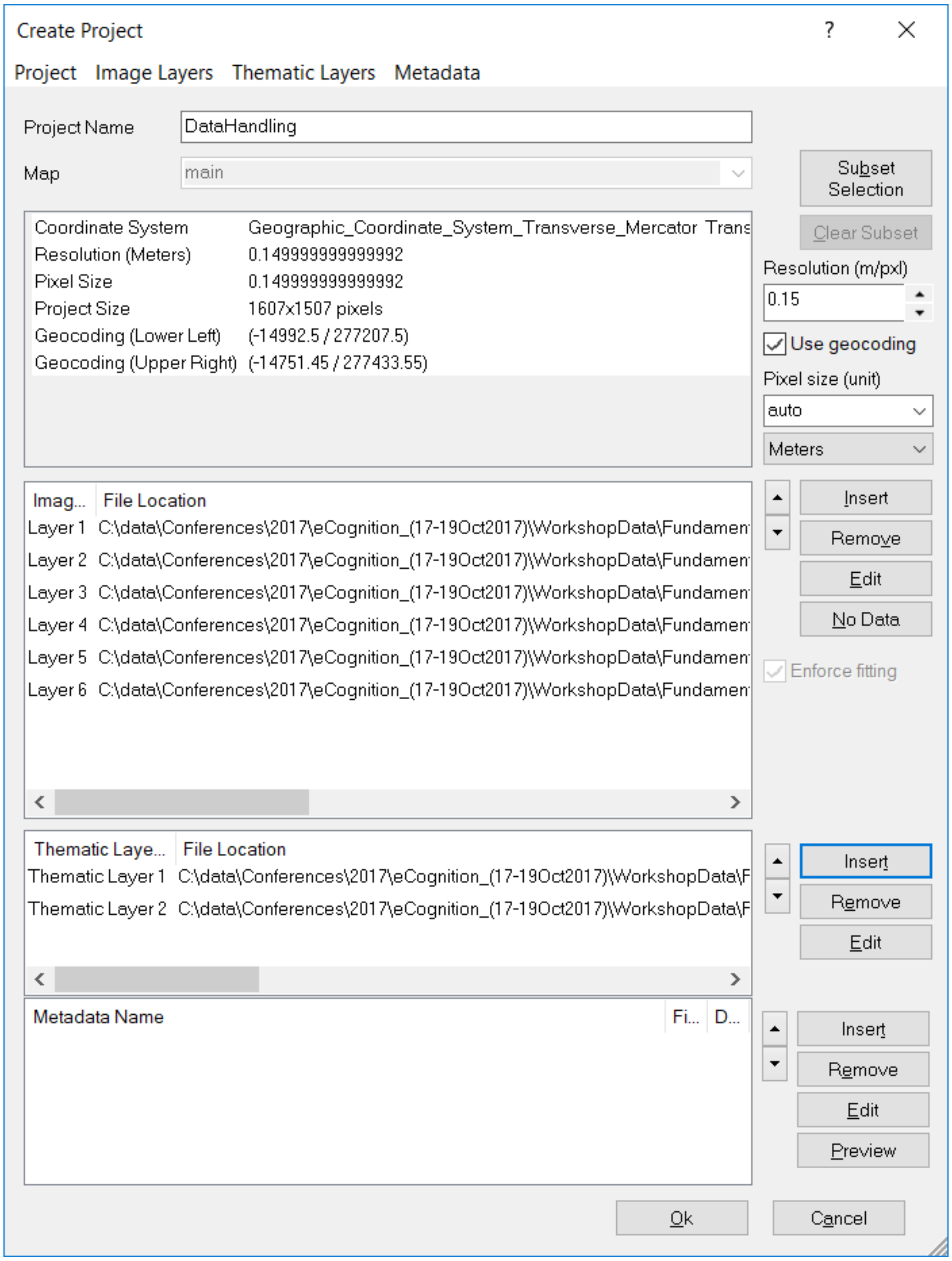
Imag...	File Location
Layer 1	C:\data\Conferences\2017\eCognition_(17-19Oct2017)\WorkshopData\Fundamen
Layer 2	C:\data\Conferences\2017\eCognition_(17-19Oct2017)\WorkshopData\Fundamen
Layer 3	C:\data\Conferences\2017\eCognition_(17-19Oct2017)\WorkshopData\Fundamen
Layer 4	C:\data\Conferences\2017\eCognition_(17-19Oct2017)\WorkshopData\Fundamen
Layer 5	C:\data\Conferences\2017\eCognition_(17-19Oct2017)\WorkshopData\Fundamen
Layer 6	C:\data\Conferences\2017\eCognition_(17-19Oct2017)\WorkshopData\Fundamen

☑ Enforce fitting

Thematic Layer Alias	Fi...	At...	W...	H...

Metadata Name	Fi...	D...

- Now adding a thematic layer from the FileGDB:



Project Name: DataHandling

Map: main

Coordinate System	Geographic_Coordinate_System_Transverse_Mercator Trans
Resolution (Meters)	0.149999999999992
Pixel Size	0.149999999999992
Project Size	1607x1507 pixels
Geocoding (Lower Left)	(-14992.5 / 277207.5)
Geocoding (Upper Right)	(-14751.45 / 277433.55)

Resolution (m/pxl): 0.15

☑ Use geocoding

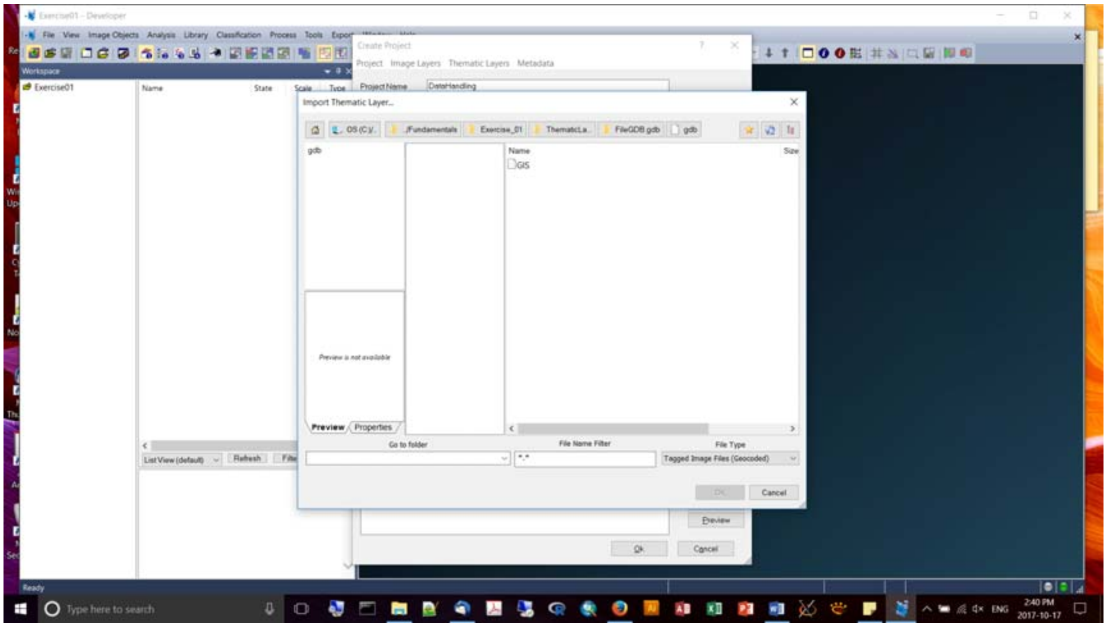
Pixel size (unit): auto

Meters

Imag...	File Location
Layer 1	C:\data\Conferences\2017\eCognition_(17-19Oct2017)\WorkshopData\Fundamen
Layer 2	C:\data\Conferences\2017\eCognition_(17-19Oct2017)\WorkshopData\Fundamen
Layer 3	C:\data\Conferences\2017\eCognition_(17-19Oct2017)\WorkshopData\Fundamen
Layer 4	C:\data\Conferences\2017\eCognition_(17-19Oct2017)\WorkshopData\Fundamen
Layer 5	C:\data\Conferences\2017\eCognition_(17-19Oct2017)\WorkshopData\Fundamen
Layer 6	C:\data\Conferences\2017\eCognition_(17-19Oct2017)\WorkshopData\Fundamen

Thematic Laye...	File Location
Thematic Layer 1	C:\data\Conferences\2017\eCognition_(17-19Oct2017)\WorkshopData\F
Thematic Layer 2	C:\data\Conferences\2017\eCognition_(17-19Oct2017)\WorkshopData\F

Metadata Name	Fi...	D...

- 

Project Name: DataHandling

Import Thematic Layer...

OS (C:) > ../Fundamentals > Exercise_01 > ThematicLa... > FileGDB.gdb > gdb

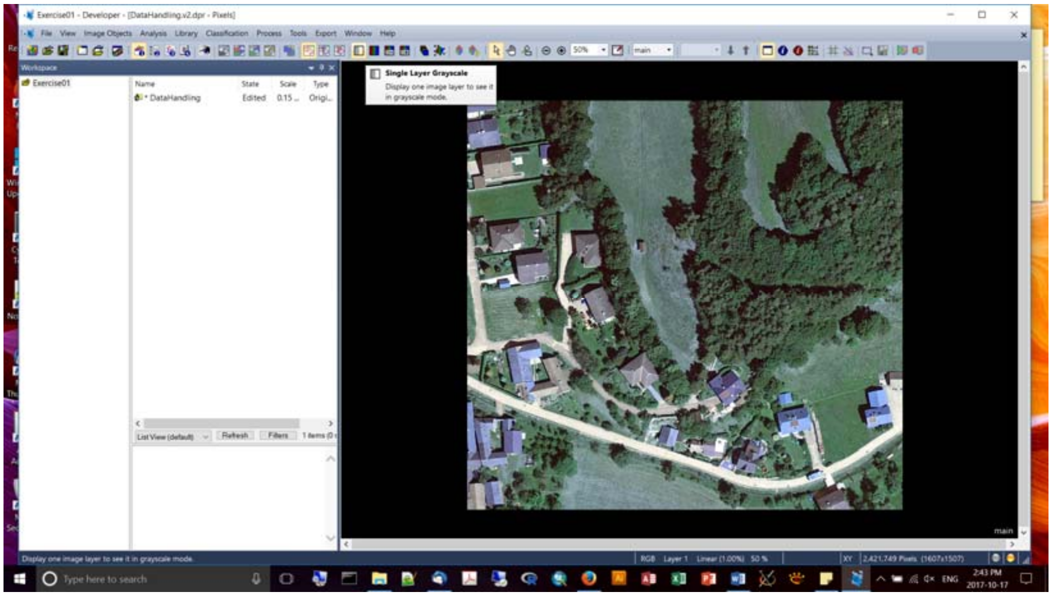
Name: GIS

File Name Filter: *.*

File Type: Tagged Image Files (Geocoded)

- Be sure to save the project! It’s not done automatically.
- Projects created with a Workspace are stored in the “dpr” sub-folder as a *.dpr file (created once the project is saved).

- Viewing tools:

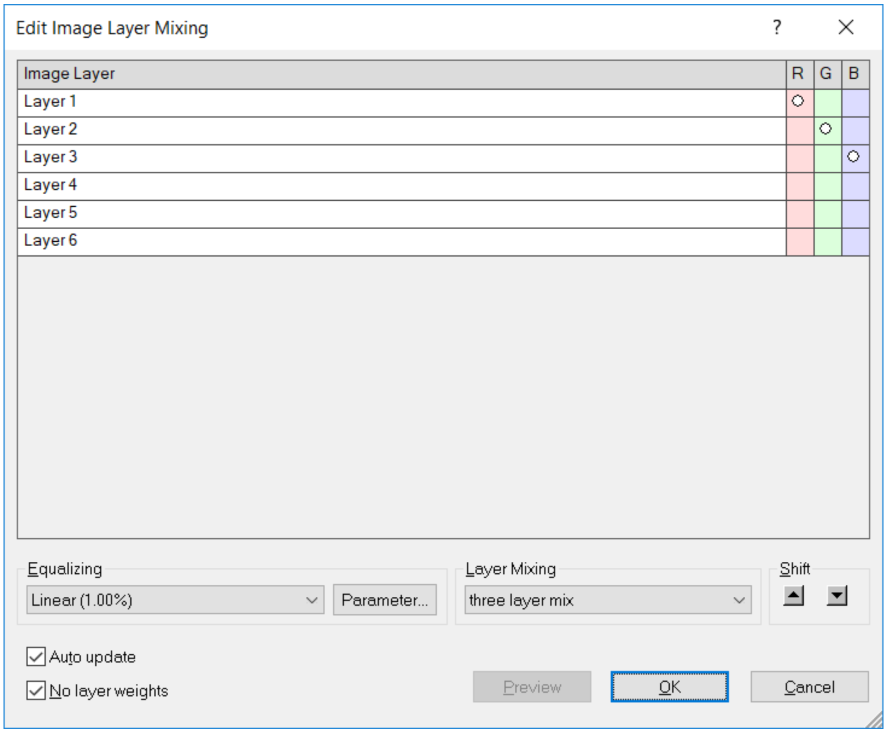


Single Layer Grayscale
Display one image layer to see it in grayscale mode.

Exercise01 - Developer - [DataHandling.v2.dpr - Pixels]
File View Image Objects Analysis Library Classification Process Tools Export Window Help
Workspace
Exercise01
Name State Scale Type
DataHandling Edited 0.15... Origi...
List View (default) Refresh Filters 1 items
50%
main
Display one image layer to see it in grayscale mode.
RGB Layer 1 Linear (1.00%) 50 %
2,421,749 Pixels (1607x1507)
Type here to search
2:43 PM
2017-10-17

- You change the way the bands are mixed (via the **Edit Image Layer Mixing** button):

- You can see that it’s good to use Layer name aliases to make them human interpretible:



Edit Image Layer Mixing

Image Layer	R	G	B
Layer 1	○		
Layer 2		○	
Layer 3			○
Layer 4			
Layer 5			
Layer 6			

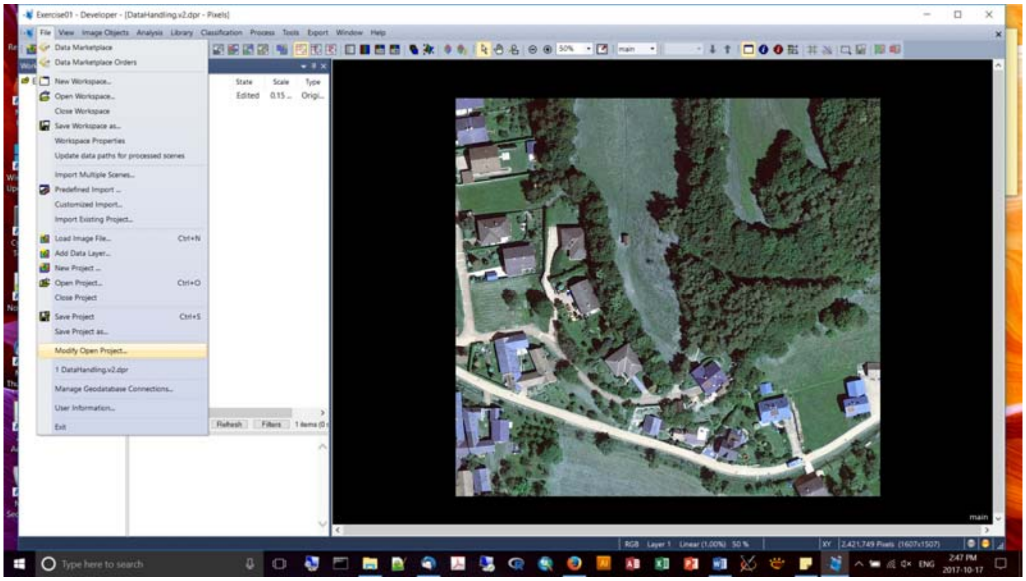
Equalizing: Linear (1.00%)

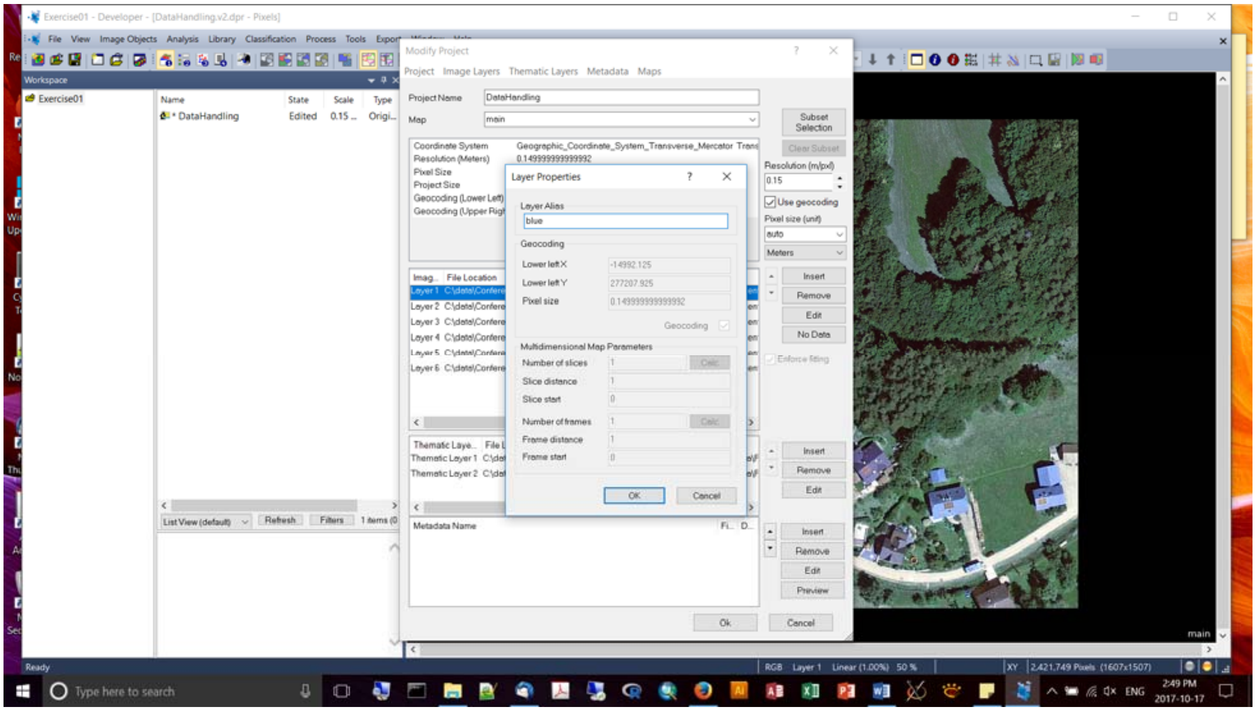
Layer Mixing: three layer mix

☑ Auto update

☑ No layer weights

- Name the layers by choosing File -> Modify Open Project





Exercise01 - Developer - [DataHandling.v2.dpr - Pixels]

Modify Project

Project Name: DataHandling

Map: main

Coordinate System	Geographic_Coordinate_System_Transverse_Mercator Trans
Resolution (Meters)	0.149999999999992
Pixel Size	
Project Size	
Geocoding (Lower Left)	
Geocoding (Upper Righ	

Resolution (m/pxl): 0.15

☑ Use geocoding

Pixel size (unit): auto

Meters

Imag...	File Location
Layer 1	C:\data\Confere
Layer 2	C:\data\Confere
Layer 3	C:\data\Confere
Layer 4	C:\data\Confere
Layer 5	C:\data\Confere
Layer 6	C:\data\Confere

Thematic Laye...	File L
Thematic Layer 1	C:\dat
Thematic Layer 2	C:\dat

Layer Properties

Layer Alias: blue

Geocoding

Lower left X	-14992.125
Lower left Y	277207.925
Pixel size	0.149999999999992

Geocoding ☑

Multidimensional Map Parameters

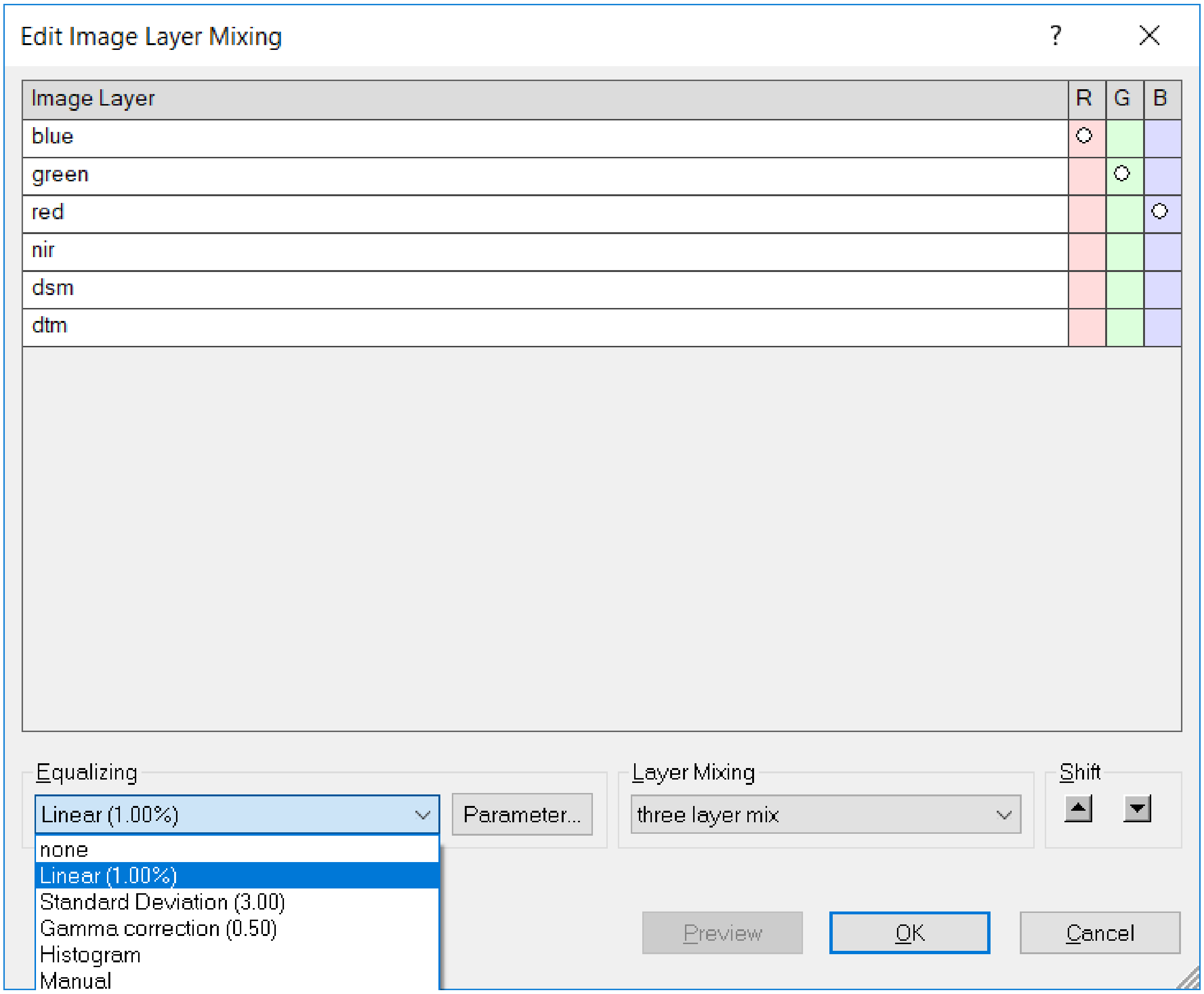
Number of slices	1
Slice distance	1
Slice start	0
Number of frames	1
Frame distance	1
Frame start	0

Name	State	Scale	Type
* DataHandling	Edited	0.15 ...	Origi...

Ready

RGB Layer 1 Linear (1.00%) 50 %

XY 2,421,749 Pixels (1607x1507)



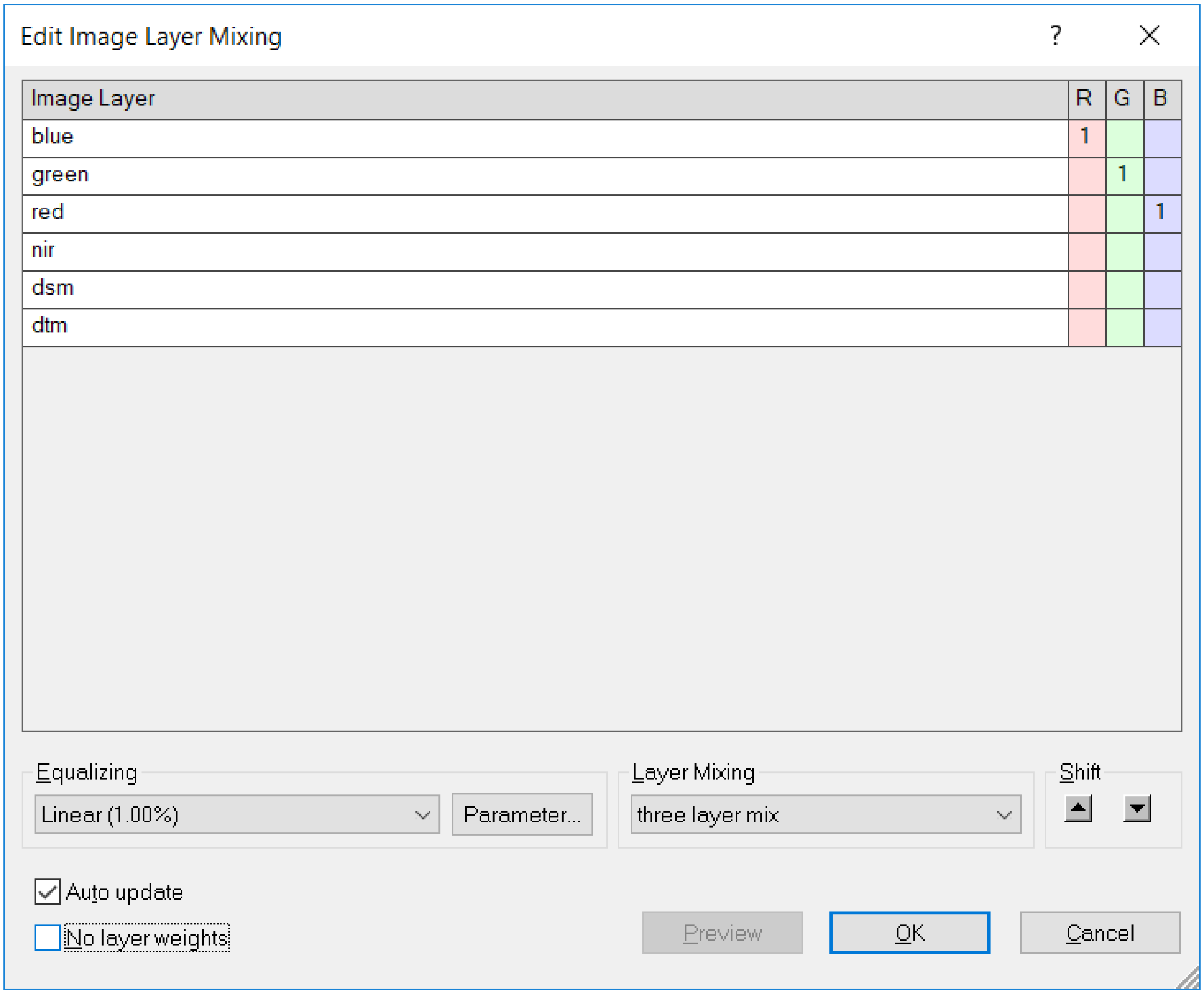
Edit Image Layer Mixing

Image Layer	R	G	B
blue	○		
green		○	
red			○
nir			
dsm			
dtm			

Equalizing: Linear (1.00%)

Layer Mixing: three layer mix

- The following demonstrates how you can access the weights by deselecting the “no layer weights” option:



Edit Image Layer Mixing

Image Layer	R	G	B
blue	1		
green		1	
red			1
nir			
dsm			
dtm			

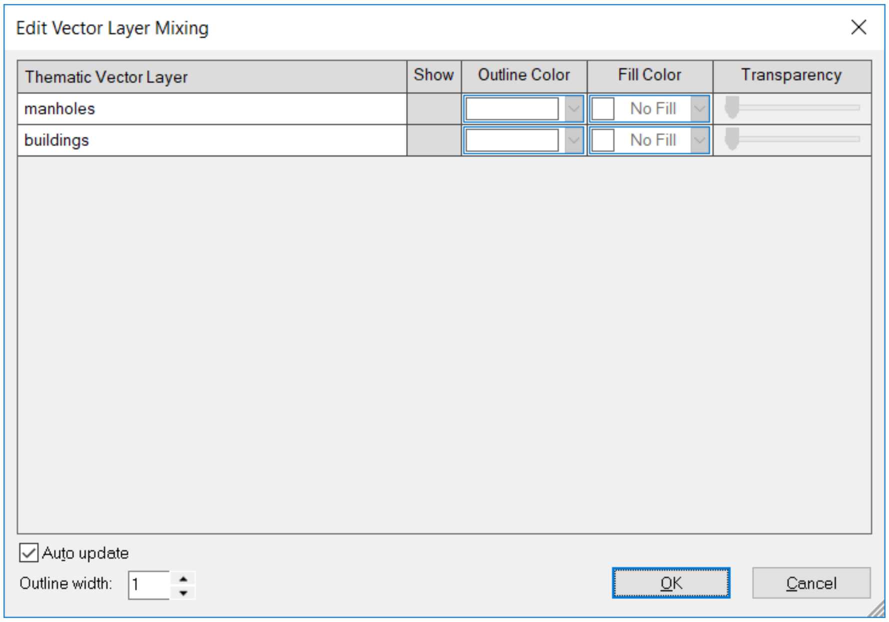
Equalizing: Linear (1.00%)

Layer Mixing: three layer mix

☑ Auto update

□ No layer weights

- Editing the vector mixing:

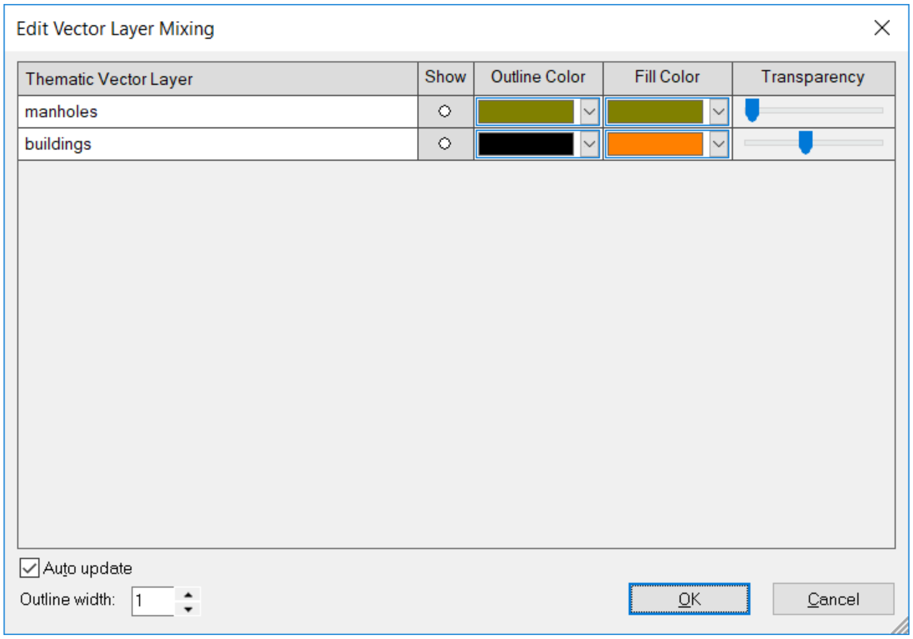


Edit Vector Layer Mixing

Thematic Vector Layer	Show	Outline Color	Fill Color	Transparency
manholes			□ No Fill	
buildings			□ No Fill	

☑ Auto update

Outline width: 1



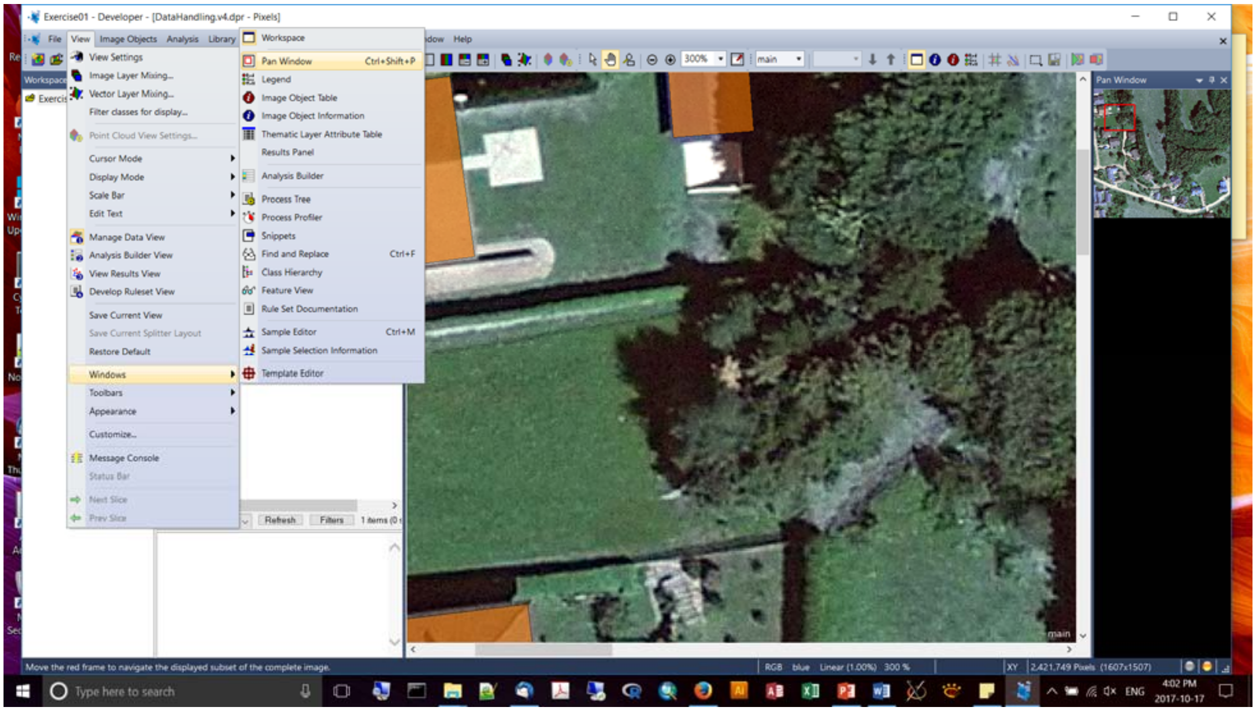
Edit Vector Layer Mixing

Thematic Vector Layer	Show	Outline Color	Fill Color	Transparency
manholes	○	🟩	🟩	
buildings	○	⬛	🟧	

☑ Auto update

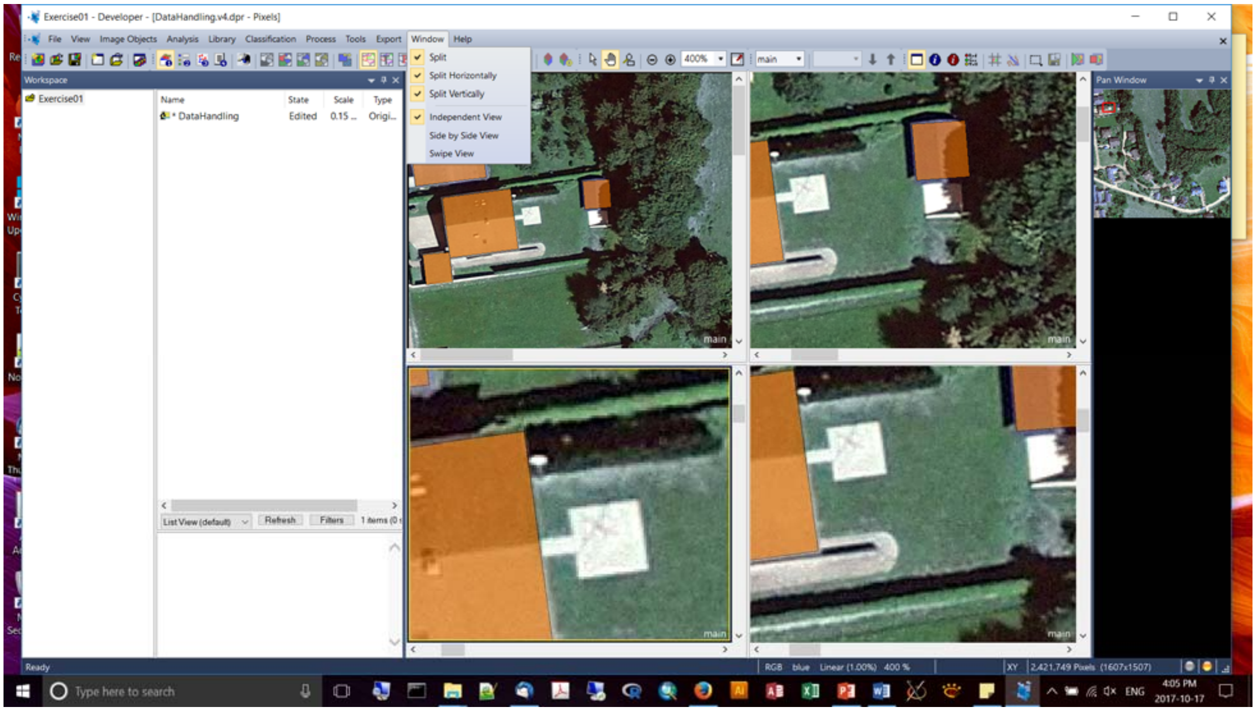
Outline width: 1

- You can open the pan window to keep track of where you are in the bigger landscape:



Move the red frame to navigate the displayed subset of the complete image.

- Splitting horizontally or vertically (via Window on main menu):



- Five processing steps:

  (1) pre-processing,

  (2) Segmentation – creates the object primitives

  (3) Classification – define classes of interest, create a class hierarchy, object classification (iterative process from 2->4; done many times)

  (4) Refinement – Can reshape objects, or establish objects of interest (iterative)

  (5) Export – Configure object features for export; export format results.

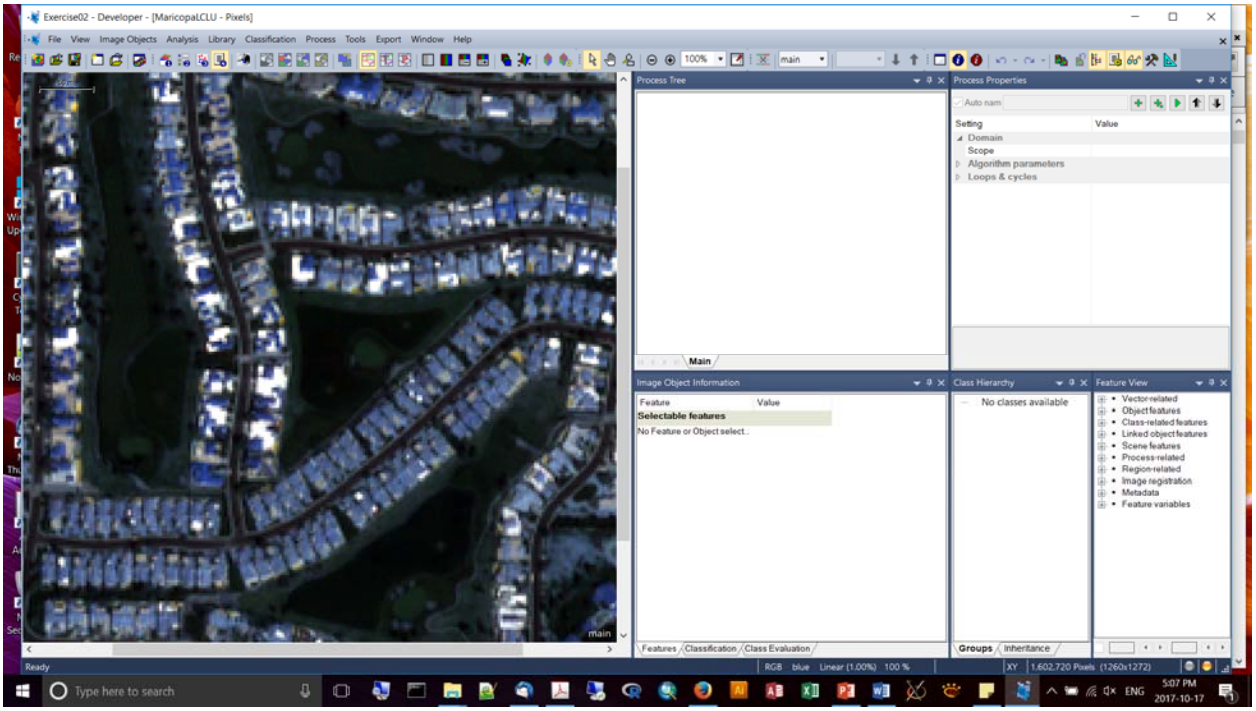
- The **Process Tree** is where you actually create and manage your processes to build a Rule set.
- The **Class Hierarchy** window allows you to create, define and manage classes
- **Image Object Information** – displays image object feature (attribute) information
- **Process Properties** – allows yo uto view/edit process parameters
- **Feature View**
- A **Proces**s is an elementary rule set procedure. Managed and executed in the Process Window tree. Processes are configured in the "Edit Process" dialog window. Processes can be arranged to form a hierarchical structure, and grouped into functional modules. You can execute all the processes, or just highlight particular sub-processes to run (important with large data sets?)
- **Domain** = region of interest to be processed (e.g., entire image)
- Types of features:
  1. Vector Features – allows addressing attributes of vector objects (e.g., perimiter of a polygon vector object)
  2. Object Features – attributes of image objects (e.g., size on an image object)
  3. Global Features

- Segmentation kicks off the initial object identification procedure. Different approaches: (1) top-down (very fast), or (2) Bottom-up (very powerful but slower).
- An object is a group of pixels and foundation of all image analyses. Objects carry features.
- Object primitives are the result of initial segmentation – usually fragments of objects of actual interest. But they carry useful information. Good object primitives are as large as possible and small enough to be used as building blocks. The image fragments can always be aggregated into bigger objects as needed.
- Projects can contain multiple image Object levels with different sized objects (scale dependent).

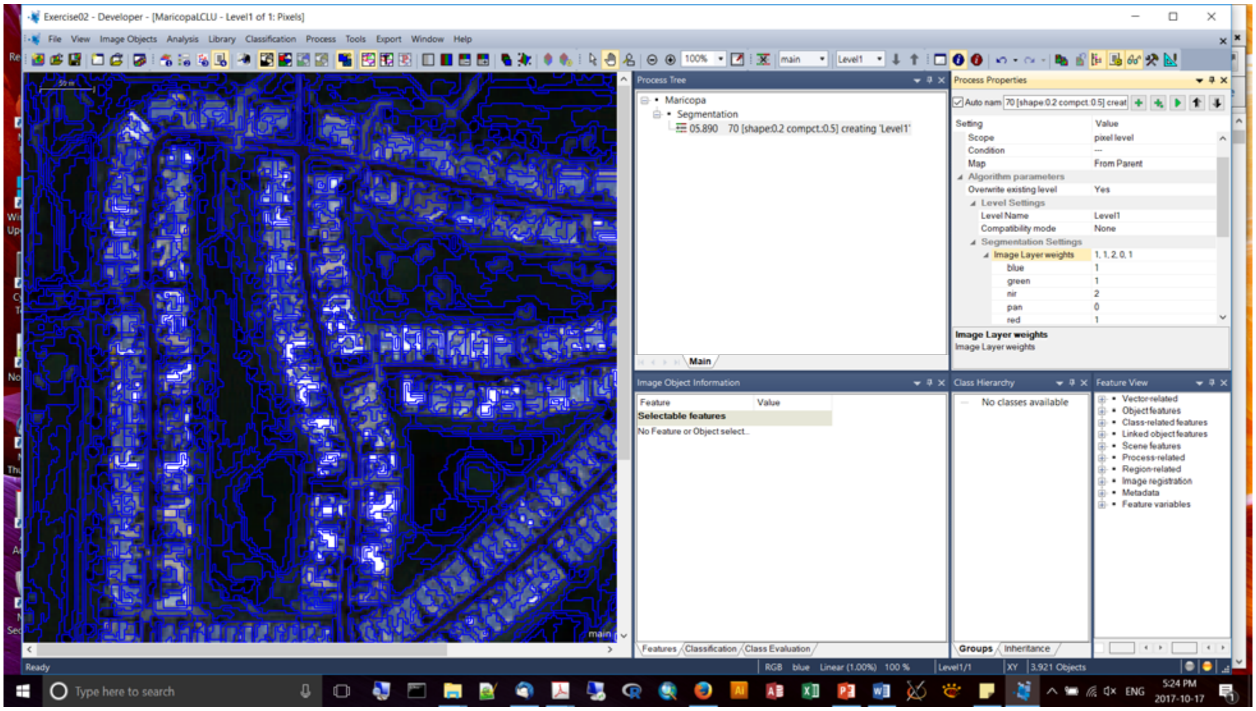
- Segmentation algorithms: (1) Chessboard, (2) Quadtree-based, (3) Multi resoluttion, (4) Multi-threshold, (5) Split
- But Segmentation also means the alteration of image objects, e.g., pixel-based object resizing, object merging, object removal, image object fusion, object growing
- **Chessboard Segmentation**: top-down approach, splits pixel or image object domain into smaller square image objects; smallest possible size is 1x1 pixel. Can be used to convert GIS vector layers to image objects (from shapefile); all cells of the same size. You can use thematic layers as "clips" to specify your AOI.
- **Quadree-based Segmentation** – top-down approach, splits pixel into a quadtree grid formed by square image objects; smallest possible size of an object is 1x1 pixel, maximum size is 256 x 256 pixels. Scale defines the maximum color difference within each selected image layer inside square image objects. Increasing scale parameter allows for more heterogeneity within the objects. It's a quick way to create object primitives. For instance – need to segregate out a large body of water? This is a quick way to segment it and set it aside.
- **Multi-resolution Segmentation** – bottom-up approach; splits pixel or image object domain based on a pairwise region merging technique; starts with single pixel-sized objects and groups like neighbours together. Main parameter is the "scale" parameter. It's the upper threshold of homogeneity. The seed looks for its best fitting neighbour for a potential merger. What scale parameter is best? No answer, depends on your sensor type, what you're trying to do, your expert opinion, etc. Look at the **ESP Tool** (Estimation of Scale Parameter) tool to help you find a good fitting scale parameter. Also: "Shape" and "compactness" are important parameters, too. Values are set between [0.1,0.9]. More compact objects are not "lengthy", i.e., linear. Layer weights can be adjusted, too, which can be quite helpful. For instance, a particular band may have more information for the object you're after.
- **Contrast-split Segmentation** – Top down approach; splits an image or image object into dark vs. bright regions. This can be a very cool way to generate the segmentation. It's incredibly powerful in many situations and is one of our favourite "go to" algorithms.
- **Multithreshold Segmentation** – bottom up approach, splits the domain into objects based on user-defined pixel threshold values. (Is this what Matt used for PIPL habitat classification?) Limited to a single image layer.

Exercise #2 – Feature Extraction

- Quickbird 5-band imagery (4 multi-spectral bands of 2.4m and one panchromatic)
- Hit #4 Develop Rulesets to display the multi-panels.
- From there, within the Process Tree window, you need to manually create a few layers in the hierarchy tree to get things started. In our case, "Maricopa", then "Segmentation" as a child process.


Exercise02 - Developer - [MaricopaLCLU - Pixels]

- Varying the scale, shape, and compactness parameters for multi-resolution segmentation:
- You can inspect the processing time on the left of the process.


Exercise02 - Developer - [MaricopaLCLU - Level1 of 1: Pixels]

Process Tree

- Maricopa
  - Segmentation
    - 05.890 70 [shape:0.2 compct.:0.5] creating 'Level1'

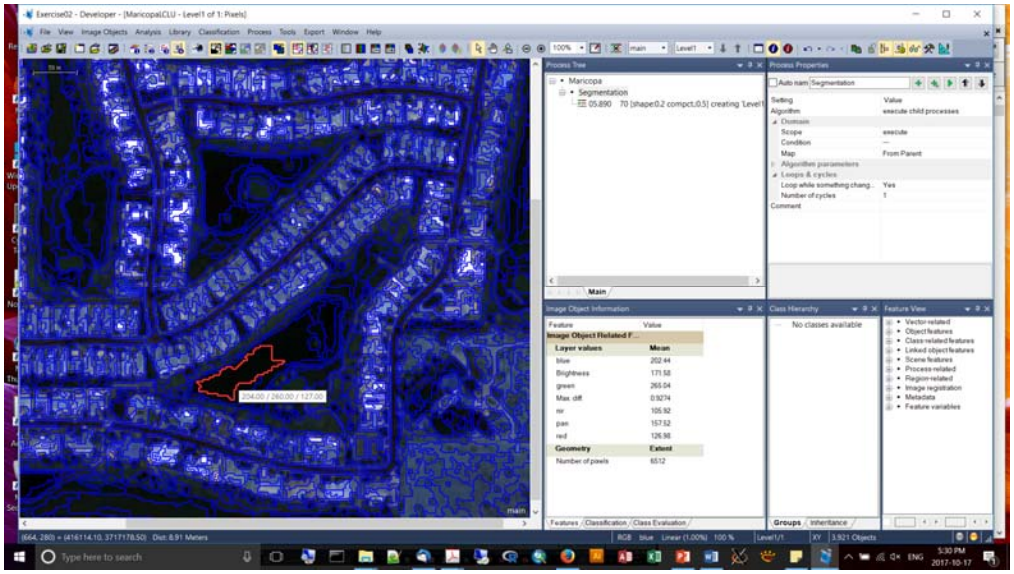
Process Properties

☑ Auto nam 70 [shape:0.2 compct.0.5] creat

Setting	Value
Scope	pixel level
Condition	---
Map	From Parent
Algorithm parameters	
Overwrite existing level	Yes
Level Settings	
Level Name	Level1
Compatibility mode	None
Segmentation Settings	
Image Layer weights	1, 1, 2, 0, 1
blue	1
green	1
nir	2
pan	0
red	1

Image Layer weights
Image Layer weights

- Querying the properties of your displayed features can be accessed from “Image Object Information” window pane – right click anywhere and select what properties you want to see:
- Pixel counts, for instance, will be in the highest resolution layer you’ve provided as input.
- Useful way to investigate the pixel values you want to use in your classification



Exercise02 - Developer - [MaricopaLCLU - Level1 of 1: Pixels]

Process Tree

- Maricopa
  - Segmentation
    - 05.890 70 [shape:0.2 compct.:0.5] creating 'Level1'

Process Properties

□ Auto nam: Segmentation

Setting	Value
Algorithm	execute child processes
Domain	
Scope	execute
Condition	—
Map	From Parent
Algorithm parameters	
Loops & cycles	
Loop while something chang...	Yes
Number of cycles	1
Comment	

Image Object Information

Feature	Value
Image Object Related F...	
Layer values	Mean
blue	202.44
Brightness	171.58
green	265.04
Max. diff.	0.9274
nir	105.92
pan	157.52
red	126.98
Geometry	Extent
Number of pixels	6512

Class Hierarchy

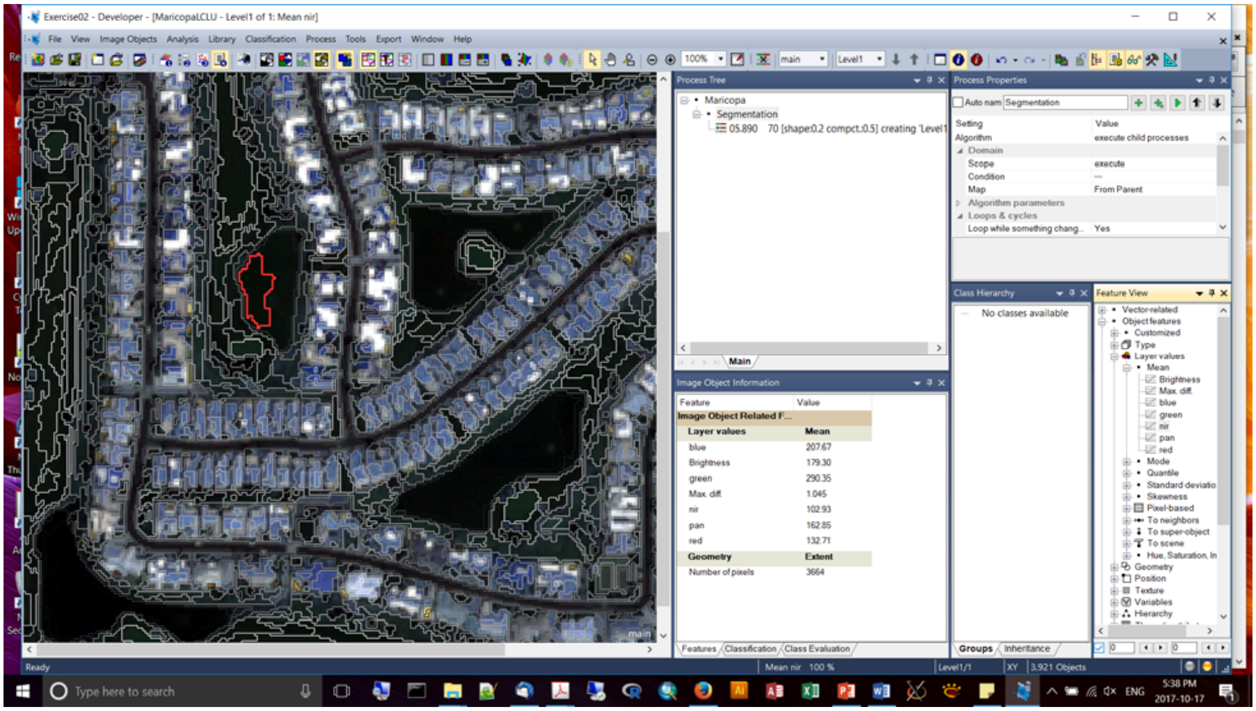
No classes available

Feature View

- Vector-related
- Object features
- Class-related features
- Linked object features
- Scene features
- Process-related
- Region-related
- Image registration
- Metadata
- Feature variables

204.00 / 260.00 / 127.00
File View Image Objects Analysis Library Classification Process Tools Export Window Help
100%
main
Level1
Main
Features Classification Class Evaluation
Groups Inheritance
(664, 280) = (416114.10, 3717178.50) Dist: 8.91 Meters
RGB blue Linear (1.00%) 100 %
Level1/1
XY
3,921 Objects
Type here to search
5:30 PM
2017-10-17

- Feature View tool – used to explore various feature values and ranges across all image objects
- If you check the box at the bottom of the Feature View Pane, you’ll be able to specify ranges of values, and you can click on different bands, e.g., NIR.



Process Tree

- Maricopa
  - Segmentation
    - 05.890 70 [shape:0.2 compct.:0.5] creating 'Level1

Main

Process Properties

☐ Auto nam Segmentation

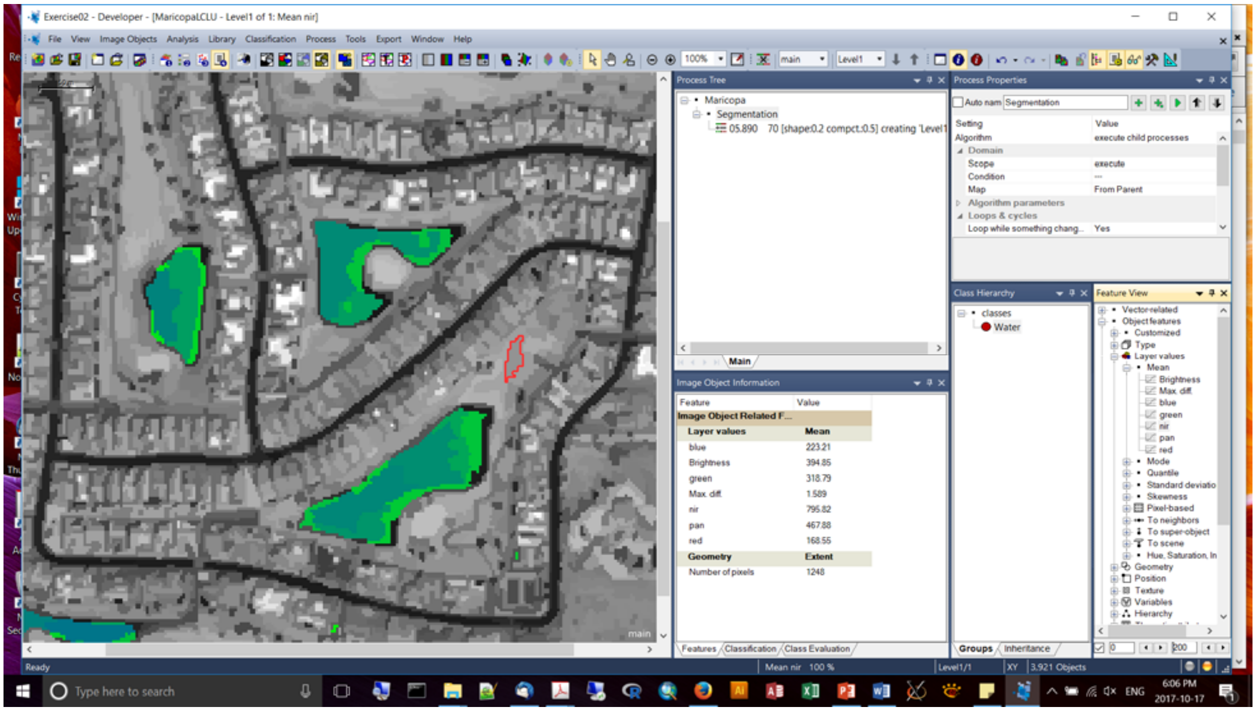
Setting	Value
Algorithm	execute child processes
Domain	
Scope	execute
Condition	---
Map	From Parent
Algorithm parameters	
Loops & cycles	
Loop while something chang...	Yes

Image Object Information

Feature	Value
Image Object Related F...	
Layer values	Mean
blue	207.67
Brightness	179.30
green	290.35
Max. diff.	1.045
nir	102.93
pan	162.85
red	132.71
Geometry	Extent
Number of pixels	3664

Features Classification Class Evaluation

- Before starting the image classification process it is helpful to start to define a class hierarchy, e.g., veg vs. none veg, for veg: forest, non forest
- Prereqs: need to have objects (from segmentation)
- Need a class description, a target class – contained within the class hierarchy
- Need a unique feature, or combination of features
- Use the Class Hierarchy window – classes can be created individually, named, have a defined color, duplicated and deleted, and can contain conditions (i.e., the class description).
- Class inheritance & semantic relationships. The class hierarchy has two tabs: (1) groups – classes grouped according to logical relationships, (2) inheritance – classes can be grouped according to their physical relationship.
- Condition from the parent class are inherited by the child class.
- “Relative border to...” --> example of a semantic class relationship. Without grouping “dense” and “sparse veg” in a group eCog won’t be able to take advantage of context. But for new users it’s fine to keep separate classes.
- Assign objects that fulfill certain criteria to a target class e.g., all dark objects are “water”. Combine multiple criteria with “and/or” operators. You can also use “soft” borders between conditions.
- Knowledge-based classification – integrate your expert knowledge of an area: (1) elevated objects with shadows; (2) islands are land surrounded by water; (3) buildings of a certain density are urban areas; (4) streets over water are bridges

- Fundamental classification algorithms: (1) “Assign Class”, and (2) “Classification” you define the thresholds.
- The following screen shot shows NIR between 0 and 200 as a good way to capture water:
- 

Exercise02 - Developer - [MaricopaLCLU - Level1 of 1: Mean nir]

Process Tree

- Maricopa
  - Segmentation
    - 05.890 70 [shape:0.2 compct.:0.5] creating 'Level1

Process Properties

Auto nam Segmentation

Setting	Value
Algorithm	execute child processes
Domain	
Scope	execute
Condition	---
Map	From Parent
Algorithm parameters	
Loops & cycles	
Loop while something chang...	Yes

Image Object Information

Feature	Value
Image Object Related F...	
Layer values	Mean
blue	223.21
Brightness	394.85
green	318.79
Max. diff.	1.589
nir	795.82
pan	467.88
red	168.55
Geometry	Extent
Number of pixels	1248

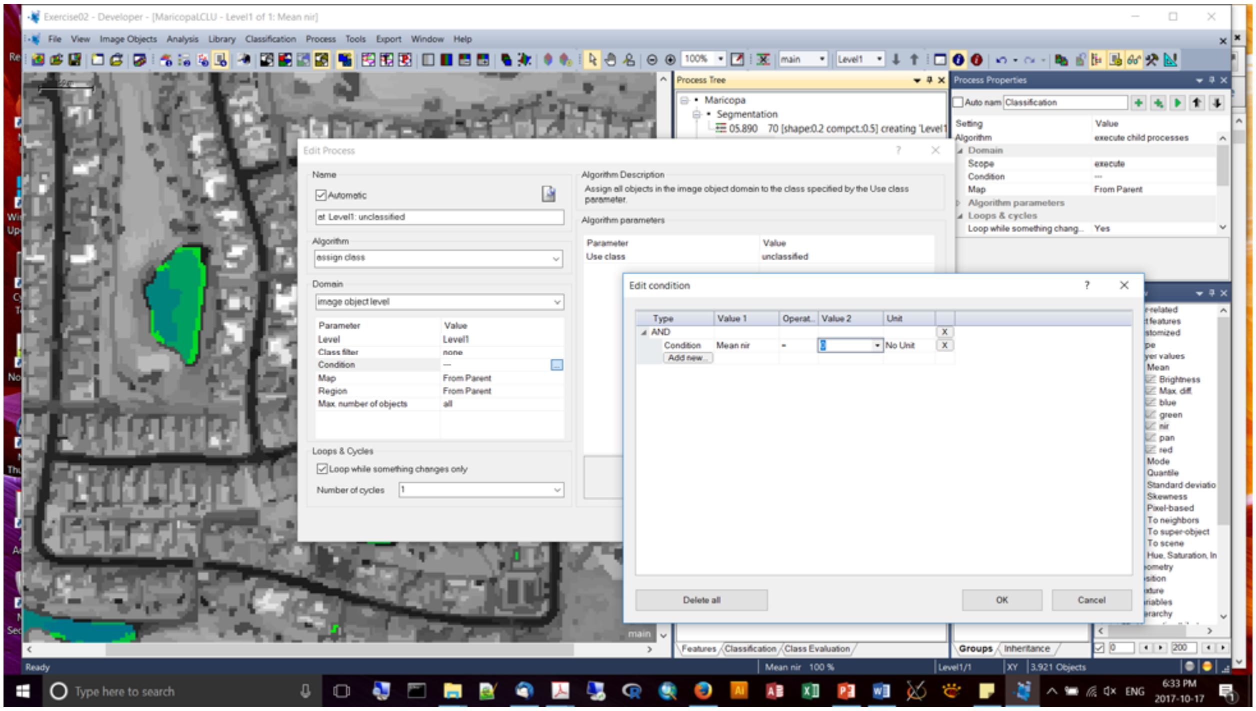
Class Hierarchy

- classes
  - Water

Feature View: 0 – 200

Ready | Mean nir 100 % | Level1/1 | XY | 3,921 Objects

- Creating conditions



Exercise02 - Developer - [MaricopaLCLU - Level1 of 1: Mean nir]

Process Tree

- Maricopa
  - Segmentation
    - 05.890 70 [shape:0.2 compct.:0.5] creating 'Level1

Process Properties

☐ Auto nam Classification

Setting	Value
Algorithm	execute child processes
Domain	
Scope	execute
Condition	---
Map	From Parent
Algorithm parameters	
Loops & cycles	
Loop while something chang...	Yes

Edit Process

Name

☑ Automatic

at Level1: unclassified

Algorithm: assign class

Domain: image object level

Parameter	Value
Level	Level1
Class filter	none
Condition	---
Map	From Parent
Region	From Parent
Max. number of objects	all

Loops & Cycles

☑ Loop while something changes only

Number of cycles: 1

Algorithm Description

Assign all objects in the image object domain to the class specified by the Use class parameter.

Algorithm parameters

Parameter	Value
Use class	unclassified

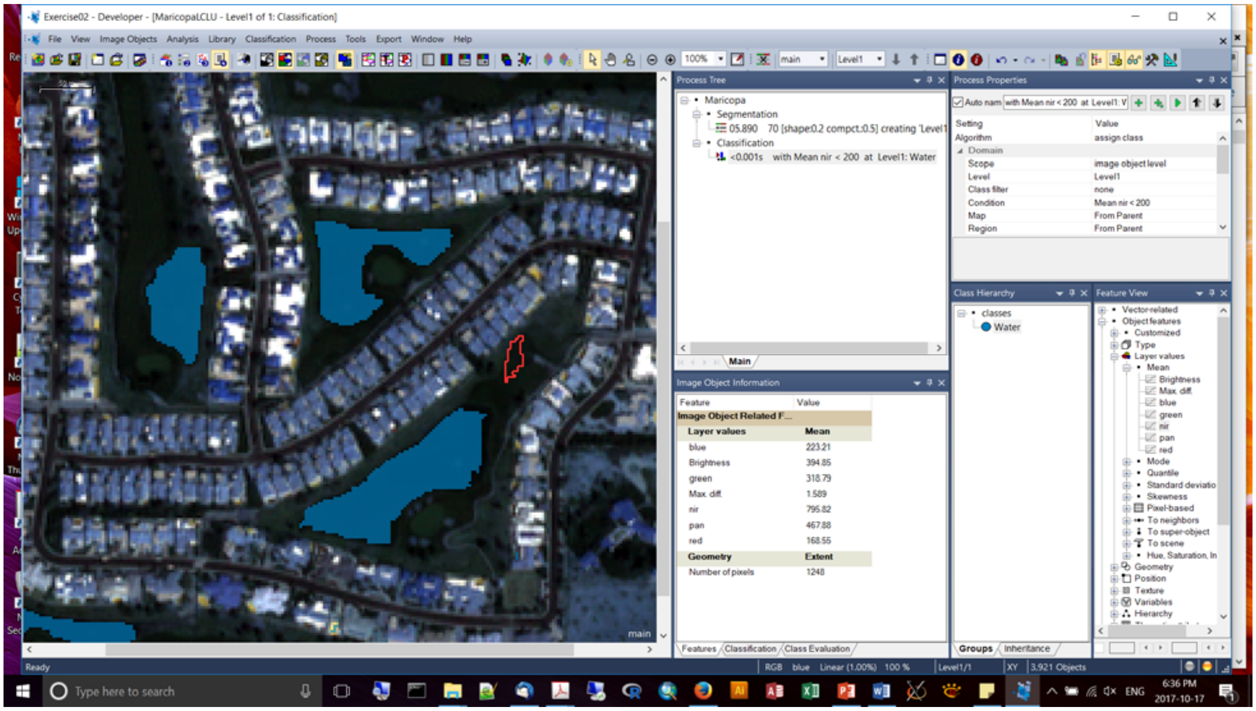
Edit condition

Type	Value 1	Operat...	Value 2	Unit
AND				
Condition	Mean nir	=	0	No Unit
Add new...				

Delete all | OK | Cancel

Ready | Mean nir 100 % | Level1/1 | XY | 3,921 Objects

- Execute your process and then hit the button "View Classification" to inspect the new classification results:



Exercise02 - Developer - [MaricopaLCLU - Level1 of 1: Classification]

Process Tree

- Maricopa
  - Segmentation
    - 05.890 70 [shape:0.2 compct.:0.5] creating 'Level1
  - Classification
    - <0.001s with Mean nir < 200 at Level1: Water

Process Properties

☑ Auto nam with Mean nir < 200 at Level1: W

Setting	Value
Algorithm	assign class
Domain	
Scope	image object level
Level	Level1
Class filter	none
Condition	Mean nir < 200
Map	From Parent
Region	From Parent

Class Hierarchy

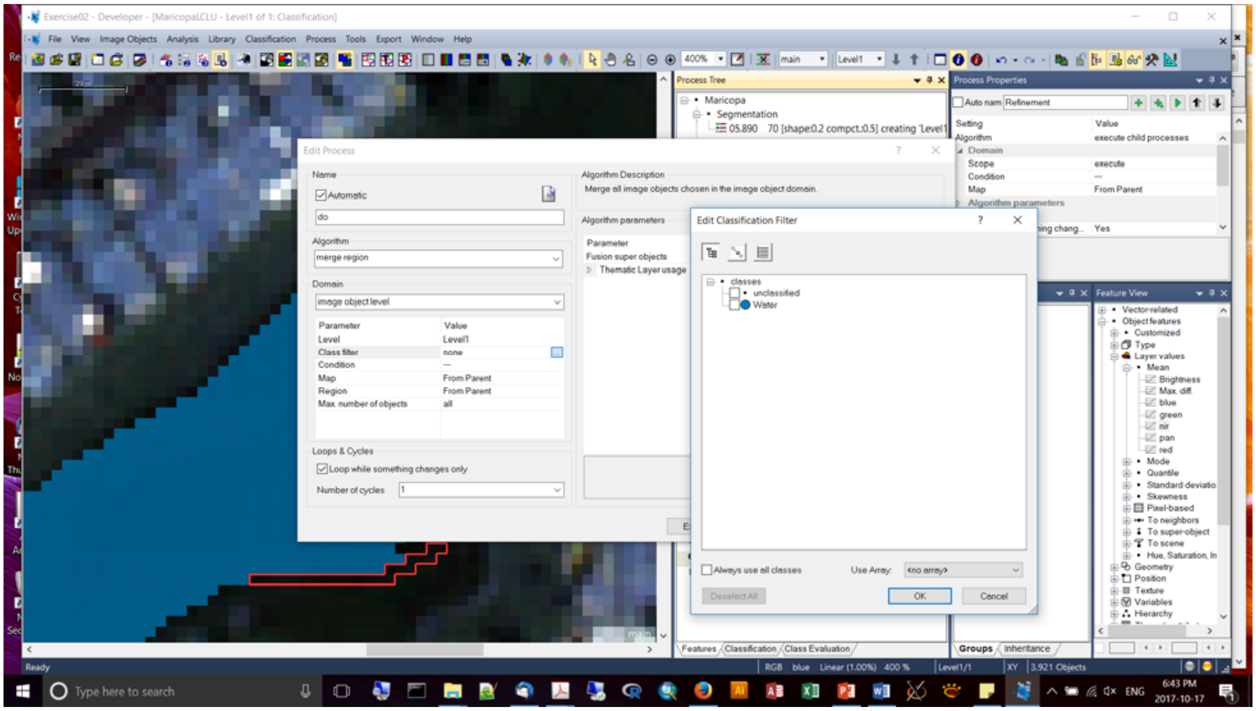
- classes
  - Water

Image Object Information

Feature	Value
Image Object Related F...	
Layer values	Mean
blue	223.21
Brightness	394.85
green	318.79
Max. diff.	1.589
nir	795.82
pan	467.88
red	168.55
Geometry	Extent
Number of pixels	1248

Ready | RGB blue Linear (1.00%) 100 % | Level1/1 | XY | 3,921 Objects

- Image Object Reshaping – we'll use an algorithm called "grow region" to grow the water into the larger boundaries.
- Then in a final step we'll remove small, trivial objects. This will eliminate potential 'salt and pepper' effects.
- Excellent for applying a minimal mapping unit.
- In this example we'll apply a merge to water:



Exercise02 - Developer - [MaricopaLCLU - Level1 of 1: Classification]

Process Tree
- Maricopa
  - Segmentation
    - 05.890 70 [shape:0.2 compct.:0.5] creating 'Level1'

Edit Process

Name: ☑ Automatic — do

Algorithm: merge region

Algorithm Description: Merge all image objects chosen in the image object domain.

Domain: image object level

Parameter	Value
Level	Level1
Class filter	none
Condition	—
Map	From Parent
Region	From Parent
Max. number of objects	all

Algorithm parameters

Parameter
Fusion super objects
Thematic Layer usage

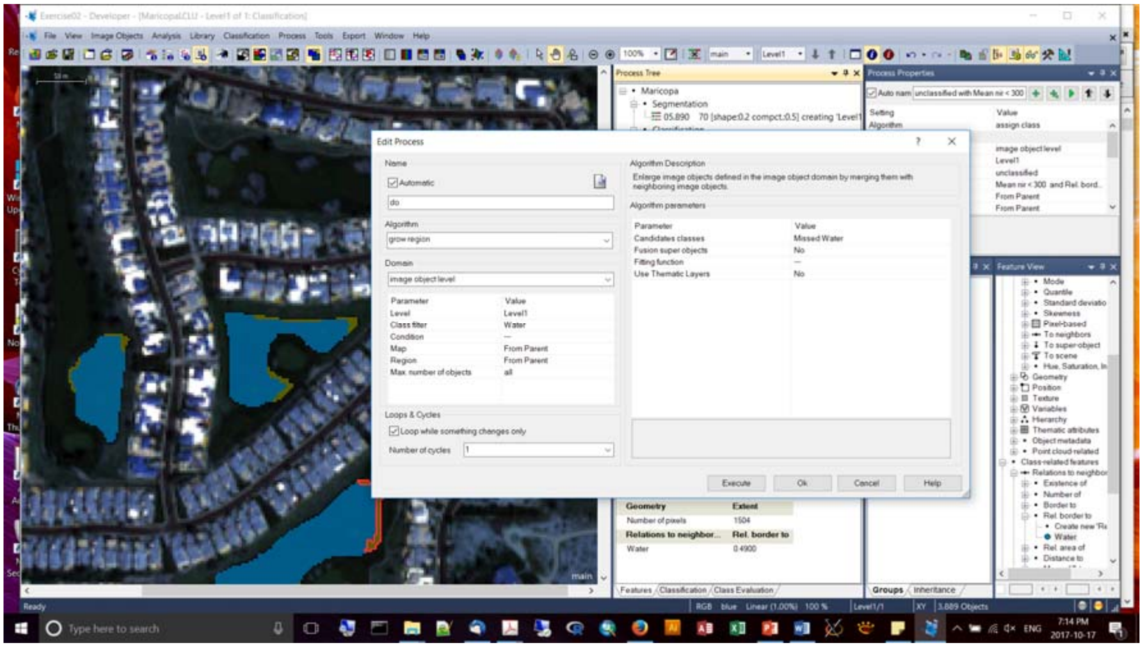
Loops & Cycles: ☑ Loop while something changes only — Number of cycles: 1

Edit Classification Filter
- classes
  - □ unclassified
  - □ Water

□ Always use all classes

Use Array: <no array>

- You can use the "grow region" to have one class blob into another class (e.g., water into "missed water"):



Edit Process

Name

☑ Automatic

do

Algorithm: grow region

Domain: image object level

Parameter	Value
Level	Level1
Class filter	Water
Condition	—
Map	From Parent
Region	From Parent
Max number of objects	all

Loops & Cycles

☑ Loop while something changes only

Number of cycles: 1

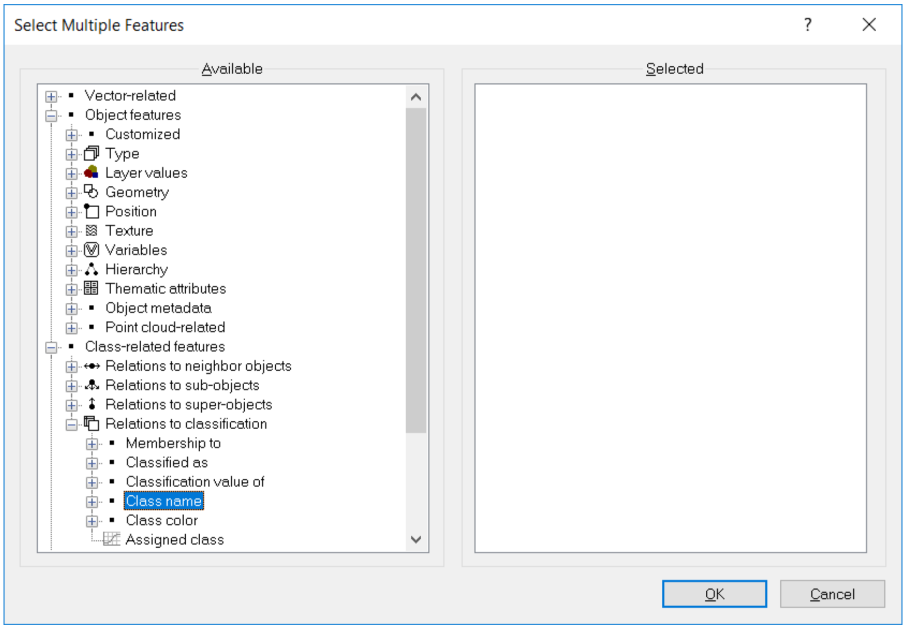
Algorithm Description

Enlarge image objects defined in the image object domain by merging them with neighboring image objects.

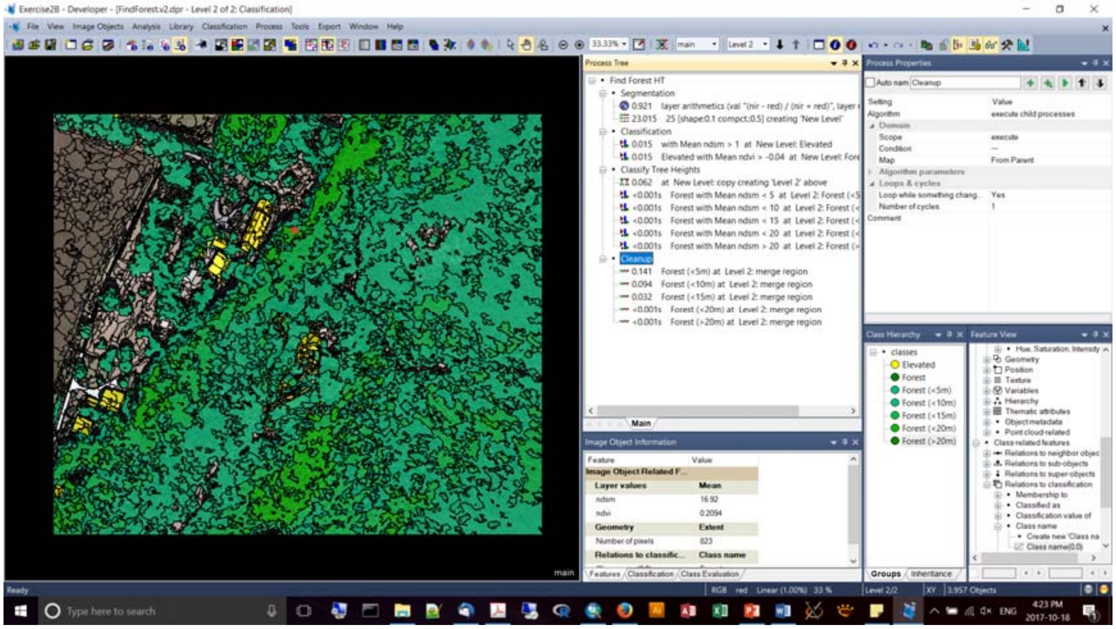
Algorithm parameters

Parameter	Value
Candidates classes	Missed Water
Fusion super objects	No
Fitting function	—
Use Thematic Layers	No

- You can export your image objects using the “export vector layer” process:
- You can specify the fields that should appear in the attribute table by clicking on the “Attributes” control and specifying which of the fields you want to appear in the export shapefile:
- How do get at the class name of the features?



- But you should first merge the features into their classes. Use the process Merge Region.



Exercise2B - Developer - [FindForest.v2.dpr - Level 2 of 2: Classification]

Process Tree

```
- Find Forest HT
  - Segmentation
      0.921   layer arithmetics (val "(nir - red) / (nir + red)", layer
      23.015   25 [shape:0.1 compct.:0.5] creating 'New Level'
  - Classification
      0.015   with Mean ndsm > 1  at  New Level: Elevated
      0.015   Elevated with Mean ndvi > -0.04  at  New Level: Fore
  - Classify Tree Heights
      0.062   at  New Level: copy creating 'Level 2' above
      <0.001s   Forest with Mean ndsm < 5  at  Level 2: Forest (<5
      <0.001s   Forest with Mean ndsm < 10  at  Level 2: Forest (<
      <0.001s   Forest with Mean ndsm < 15  at  Level 2: Forest (<
      <0.001s   Forest with Mean ndsm < 20  at  Level 2: Forest (<
      <0.001s   Forest with Mean ndsm > 20  at  Level 2: Forest (>
  - Cleanup
      0.141   Forest (<5m) at  Level 2: merge region
      0.094   Forest (<10m) at  Level 2: merge region
      0.032   Forest (<15m) at  Level 2: merge region
      <0.001s   Forest (<20m) at  Level 2: merge region
      <0.001s   Forest (>20m) at  Level 2: merge region
```

Process Properties

□ Auto nam Cleanup

Setting	Value
Algorithm	execute child processes
Domain	
Scope	execute
Condition	—
Map	From Parent
Algorithm parameters	
Loops & cycles	
Loop while something chang...	Yes
Number of cycles	1
Comment	

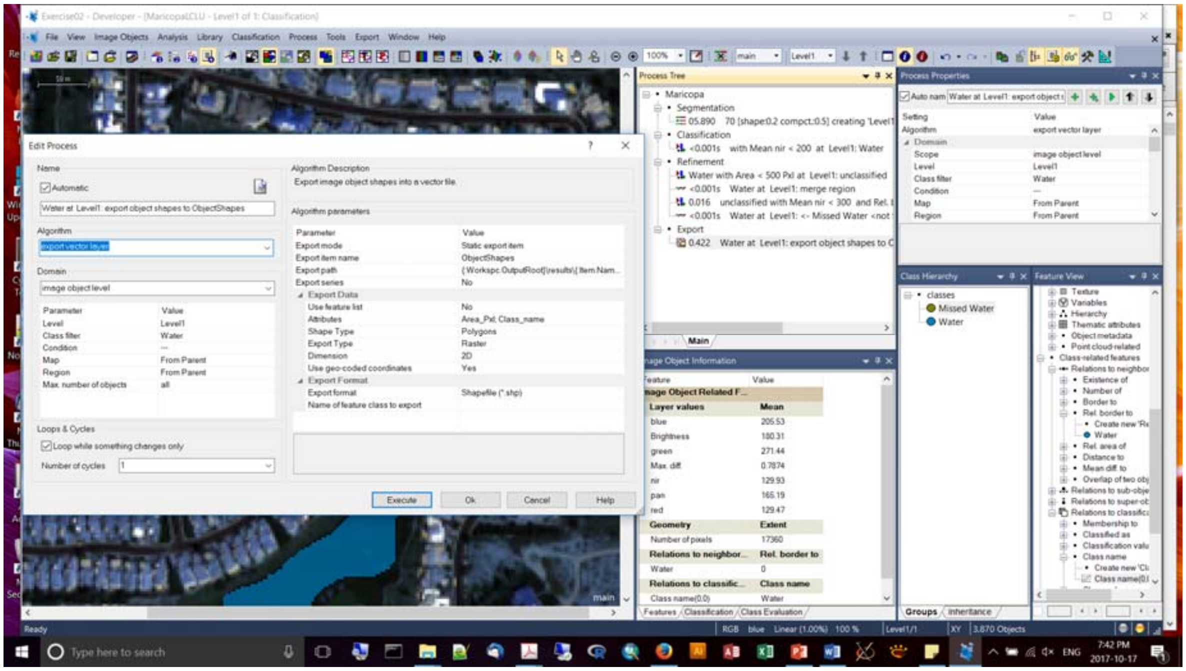
Class Hierarchy

- classes
  - ● Elevated
  - ● Forest
  - ● Forest (<5m)
  - ● Forest (<10m)
  - ● Forest (<15m)
  - ● Forest (<20m)
  - ● Forest (>20m)

Image Object Information

Feature	Value
Image Object Related F...	
Layer values	**Mean**
ndsm	16.92
ndvi	0.2094
Geometry	**Extent**
Number of pixels	823
Relations to classific...	**Class name**

File View Image Objects Analysis Library Classification Process Tools Export Window Help
33.33%
main
Level 2
Main
Features Classification Class Evaluation
Groups Inheritance
Feature View
Ready
RGB red Linear (1.00%) 33 %
Level 2/2
XY 3,957 Objects
Type here to search
4:23 PM
2017-10-18



Exercise02 - Developer - [MaricopaLCLU - Level1 of 1: Classification]

Process Tree

```
- Maricopa
  - Segmentation
      05.890   70 [shape:0.2 compct.:0.5] creating 'Level1'
  - Classification
      <0.001s   with Mean nir < 200  at  Level1: Water
  - Refinement
      Water with Area < 500 Pxl at  Level1: unclassified
      <0.001s   Water at  Level1: merge region
      0.016   unclassified with Mean nir < 300  and Rel. b
      <0.001s   Water at  Level1: <- Missed Water <not
  - Export
      0.422   Water at  Level1: export object shapes to O
```

Edit Process

Name

☑ Automatic

Water at Level1: export object shapes to ObjectShapes

Algorithm: export vector layer

Domain: image object level

Parameter	Value
Level	Level1
Class filter	Water
Condition	—
Map	From Parent
Region	From Parent
Max. number of objects	all

Loops & Cycles

☑ Loop while something changes only

Number of cycles: 1

Algorithm Description

Export image object shapes into a vector file

Algorithm parameters

Parameter	Value
Export mode	Static export item
Export item name	ObjectShapes
Export path	{:Workspc.OutputRoot}\results\{:Item.Nam...
Export series	No
Export Data	
Use feature list	No
Attributes	Area_Pxl; Class_name
Shape Type	Polygons
Export Type	Raster
Dimension	2D
Use geo-coded coordinates	Yes
Export Format	
Export format	Shapefile (*.shp)
Name of feature class to export	

Process Properties

☑ Auto nam Water at Level1: export object s

Setting	Value
Algorithm	export vector layer
Domain	
Scope	image object level
Level	Level1
Class filter	Water
Condition	—
Map	From Parent
Region	From Parent

Class Hierarchy

- classes
  - ● Missed Water
  - ● Water

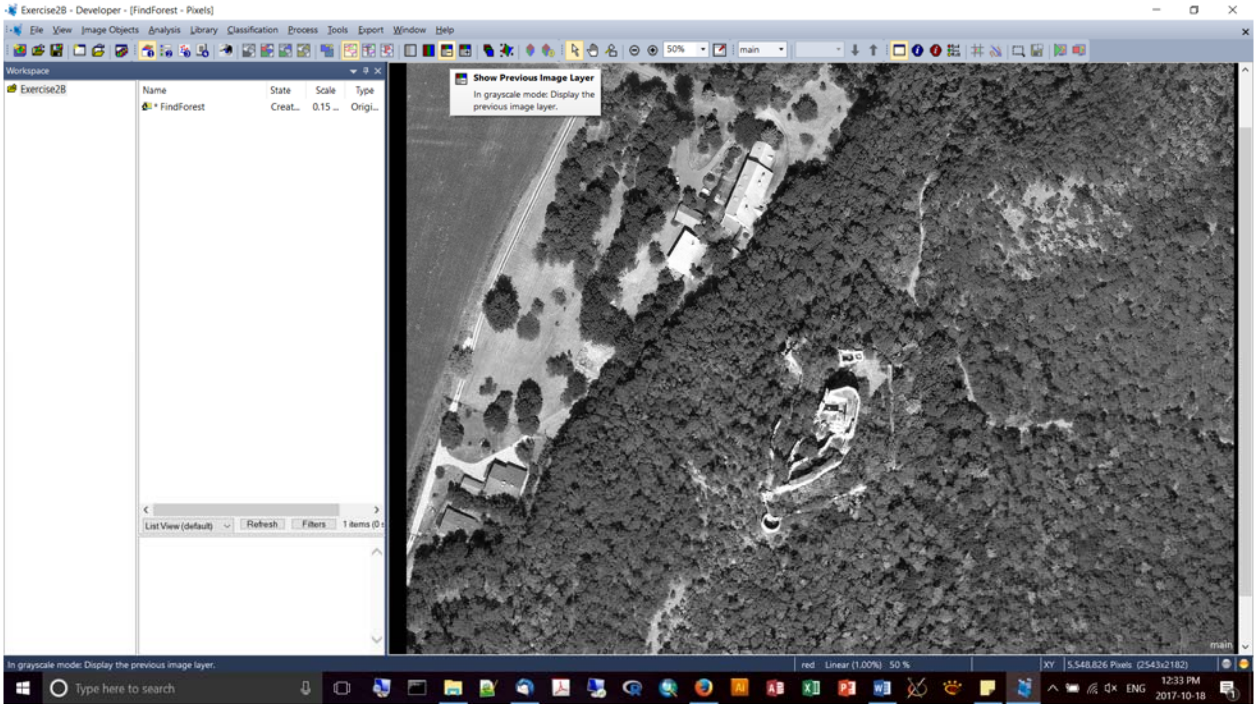
Image Object Information

Feature	Value
Image Object Related F...	
Layer values	**Mean**
blue	205.53
Brightness	180.31
green	271.44
Max. diff	0.7874
nir	129.93
pan	165.19
red	129.47
Geometry	**Extent**
Number of pixels	17360
Relations to neighbor...	**Rel. border to**
Water	0
Relations to classific...	**Class name**
Class name(0,0)	Water

File View Image Objects Analysis Library Classification Process Tools Export Window Help
100%
main
Level1
Execute
Ok
Cancel
Help
Main
Features Classification Class Evaluation
Groups Inheritance
Feature View
Ready
RGB blue Linear (1.00%) 100 %
Level1/1
XY 3,870 Objects
Type here to search
7:42 PM
2017-10-17

OCT.18, 2017 (Day #2)

- In the Ex #2b, you’ll be working with an **ndsm**, which is a normalized digital surface model (=dsm – dtm). DTM = bare earth. DSM = surface model, including above ground structures/features.
- Note how you can toggle through your data layers using the “next layer”, ‘previous layer” buttons.



Exercise2B - Developer - [FindForest - Pixels]

Show Previous Image Layer

In grayscale mode: Display the previous image layer.

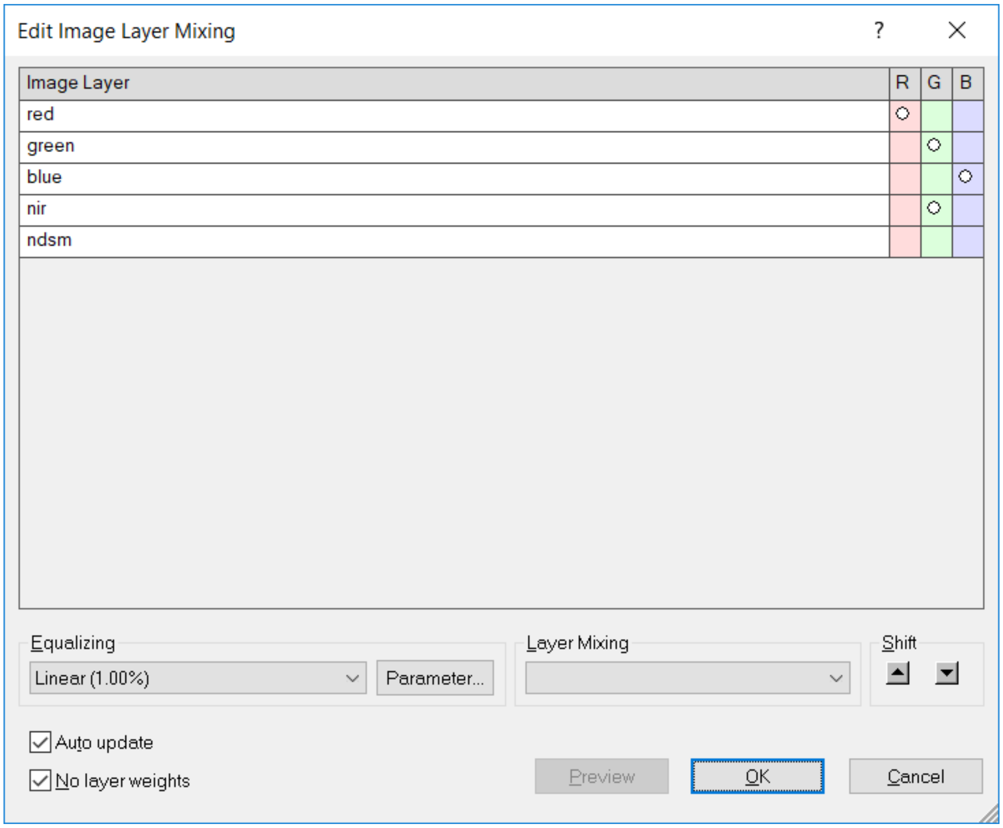
Name	State	Scale	Type
* FindForest	Creat...	0.15 ...	Origi...

In grayscale mode: Display the previous image layer.

red Linear (1.00%) 50 %

XY 5,548,826 Pixels (2543x2182)

- Experimenting with blending of bands to enhance feature exploration:
- In this case, NIR has been assigned to the green band.



Edit Image Layer Mixing

Image Layer	R	G	B
red	○		
green		○	
blue			○
nir		○	
ndsm			

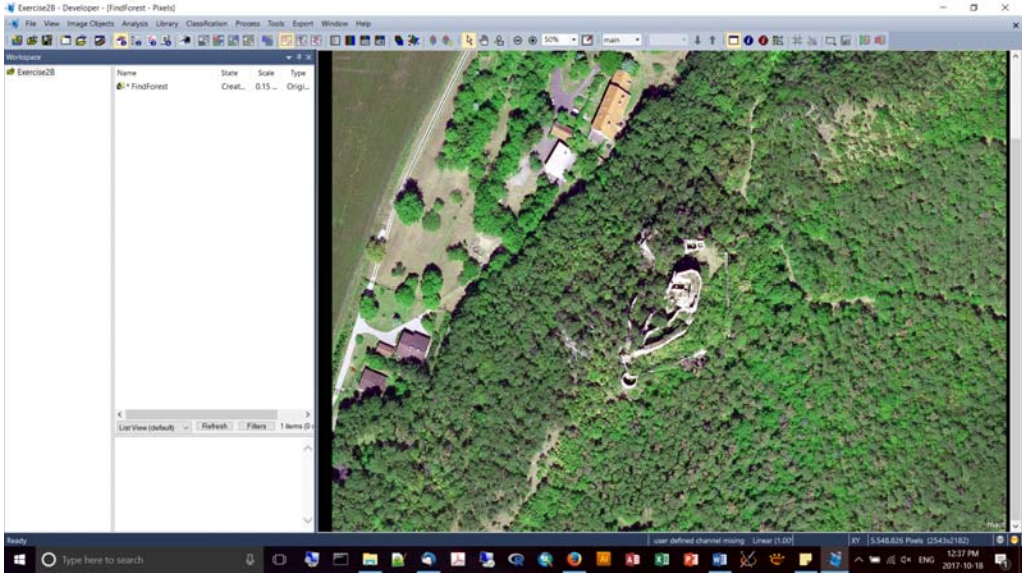
Equalizing: Linear (1.00%)

Layer Mixing:

☑ Auto update

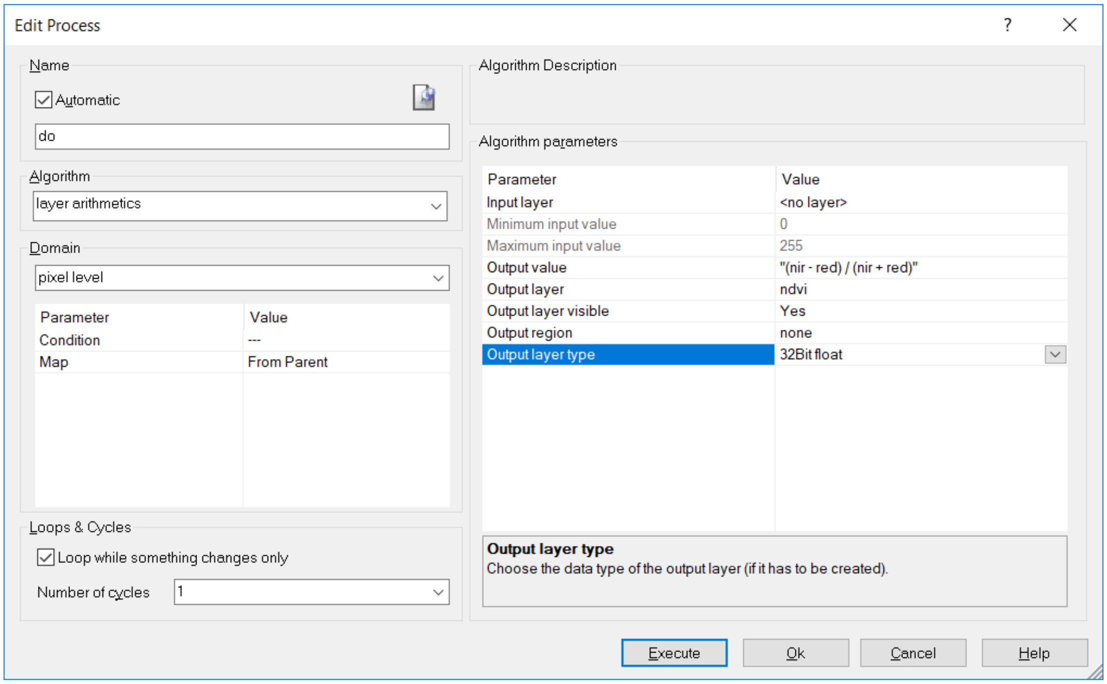
☑ No layer weights

- Notice the way the vegetation really stands out:
- But also, you can start to distinguish different species (conifers vs. non-conifers)



Exercise28 - Developer - [FindForest - Pixels]
File View Image Objects Analysis Library Classification Process Tools Export Window Help
Workspace
Exercise28
Name State Scale Type
FindForest Creat... 0.15... Origi...
List View (default) Refresh Filters
user defined channel mixing Linear (1.00)
Type here to search

- You can do some pre-processing within eCog, e.g., NDVI calculation.
- There is an algorithm to do raster calculation (“layer arithmetics”)
- Multi-resolution segmentation = one of the strongest for natural environments.
- E.g.,



Name

☑ Automatic

do

Algorithm

layer arithmetics

Domain

pixel level

Parameter	Value
Condition	---
Map	From Parent

Loops & Cycles

☑ Loop while something changes only

Number of cycles 1

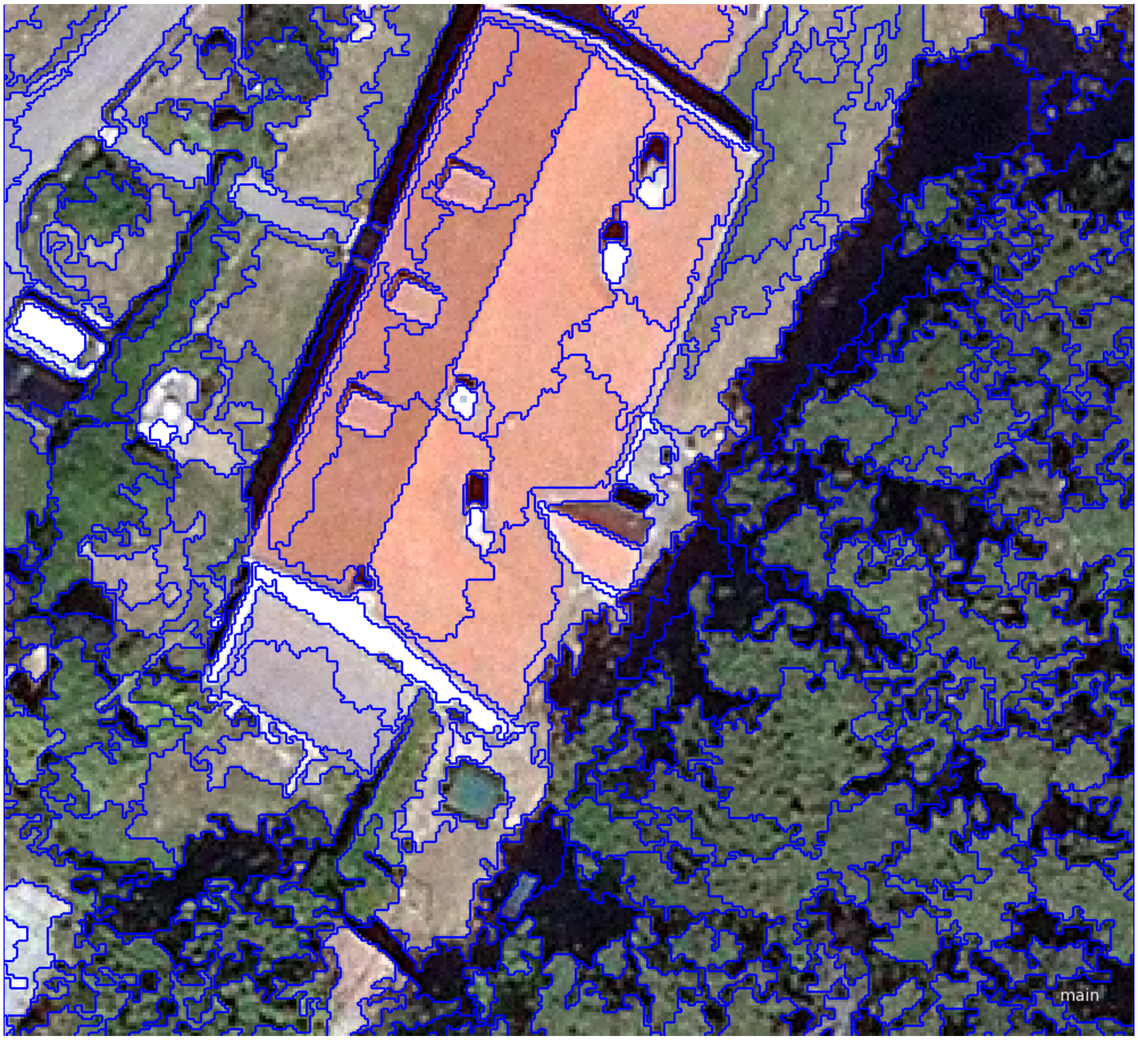
Algorithm Description

Algorithm parameters

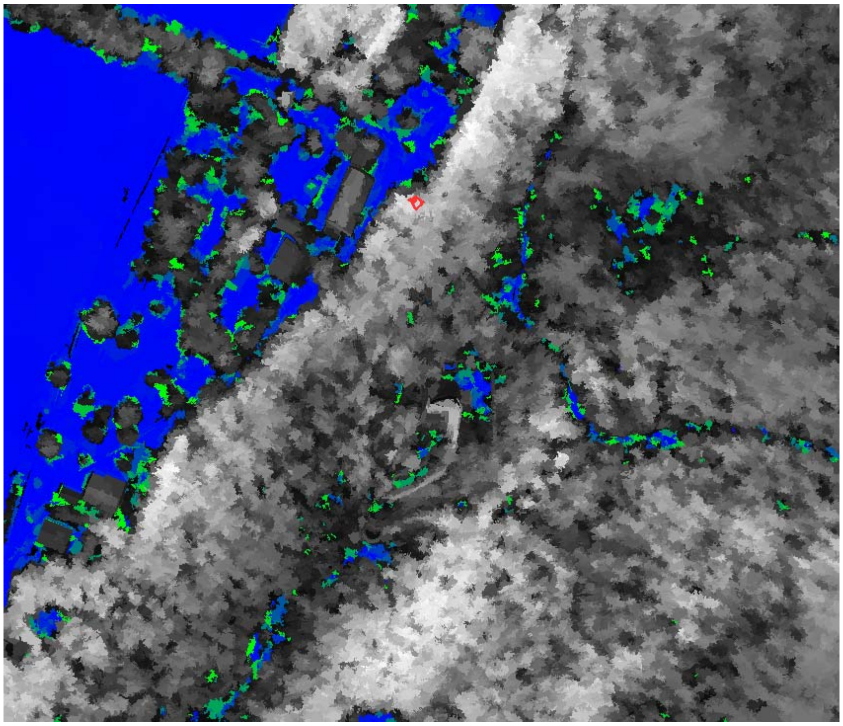
Parameter	Value
Input layer	<no layer>
Minimum input value	0
Maximum input value	255
Output value	"(nir - red) / (nir + red)"
Output layer	ndvi
Output layer visible	Yes
Output region	none
Output layer type	32Bit float

Output layer type
Choose the data type of the output layer (if it has to be created).

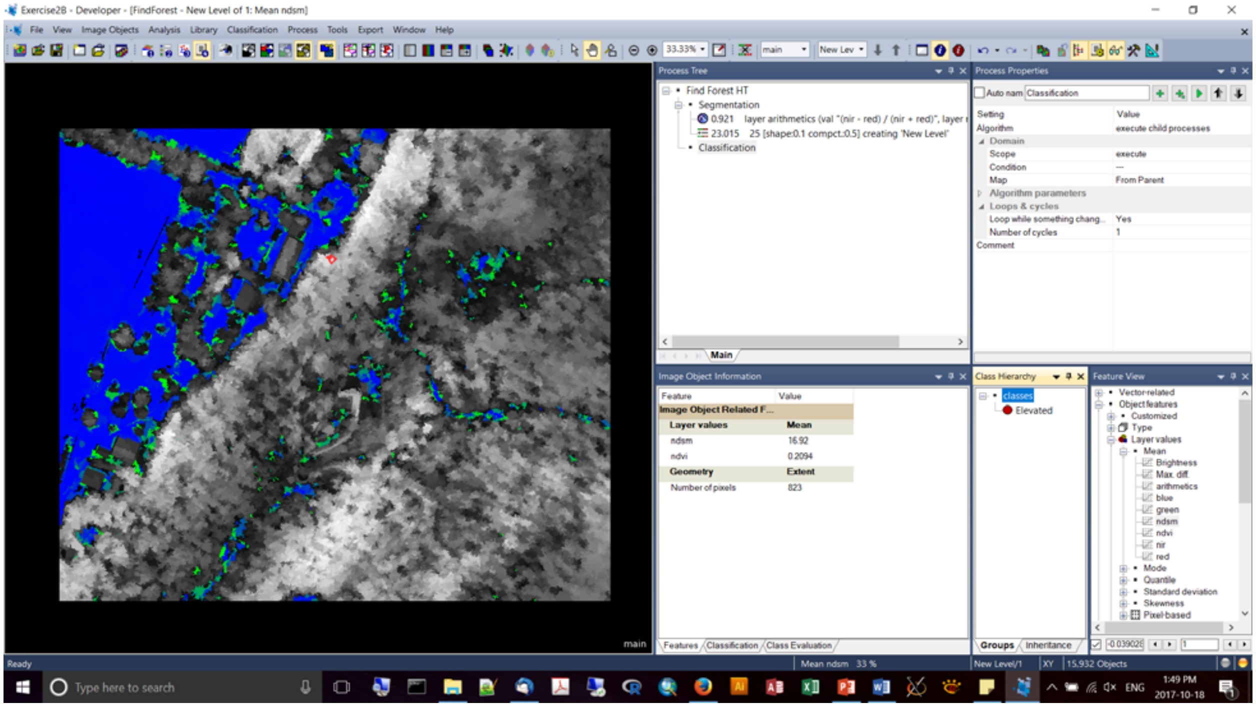
- This did a good job of distinguishing forest from shadow; the builidng is pretty well identified.



- For this project, we're going to employ a hierarchical classification method.
- Start by separating forest v non-forest. What's unique about forest? They're elevated and "green" (i.e., high NDVI).
- Hierarchy: Elev vs. Non-Elev.
- Elev: Vegetated vs. Non-Vegetated.
- <u>The ndsm will provide the elevation information of above-ground features</u>. No terrain! Just above ground stuff like trees and buildings. --> inspection of features in "Feature view" suggest a possible cutoff for height is 1m. Elevated objects will have ndsm > 1. To execute we need a class, a feature, a threshold, and an algorithm.



- N.B. Before we can classify (assign class) we need to first create the class in the “Class Hierarchy window”:



Exercise2B - Developer - [FindForest - New Level of 1: Mean ndsm]

Process Tree

```
Find Forest HT
    Segmentation
        0.921   layer arithmetics (val "(nir - red) / (nir + red)", layer
        23.015   25 [shape:0.1 compct.:0.5] creating 'New Level'
    Classification
```

Process Properties

☐ Auto nam Classification

Setting	Value
Algorithm	execute child processes
Domain	
Scope	execute
Condition	---
Map	From Parent
Algorithm parameters	
Loops & cycles	
Loop while something chang...	Yes
Number of cycles	1
Comment	

Image Object Information

Feature	Value
Image Object Related F...	
Layer values	Mean
ndsm	16.92
ndvi	0.2094
Geometry	Extent
Number of pixels	823

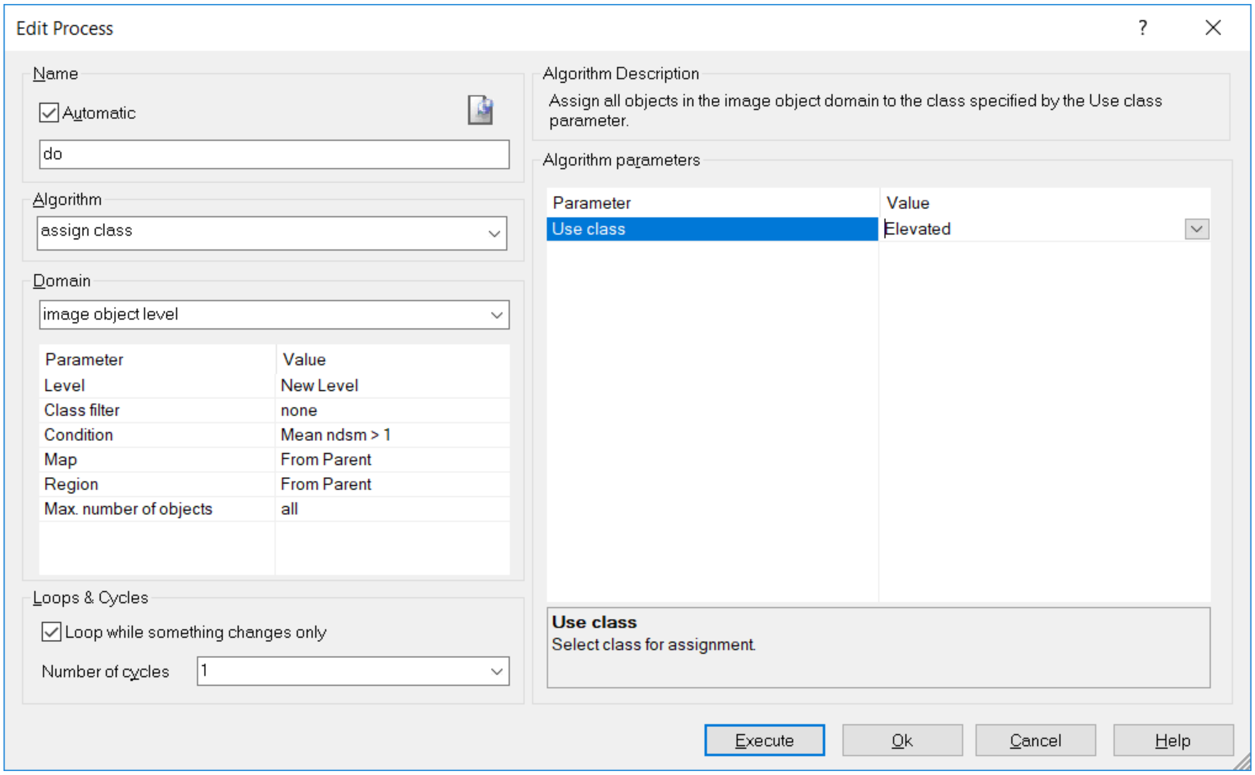
Class Hierarchy

```
classes
    Elevated
```

Feature View

```
Vector-related
Object features
    Customized
    Type
    Layer values
        Mean
            Brightness
            Max. diff.
            arithmetics
            blue
            green
            ndsm
            ndvi
            nir
            red
        Mode
        Quantile
        Standard deviation
        Skewness
        Pixel-based
```

- Notice how we choose the class that will be assigned ("Elevated") on the right side.



Edit Process

Name

☑ Automatic

do

Algorithm

assign class

Domain

image object level

Parameter	Value
Level	New Level
Class filter	none
Condition	Mean ndsm > 1
Map	From Parent
Region	From Parent
Max. number of objects	all

Loops & Cycles

☑ Loop while something changes only

Number of cycles 1

Algorithm Description

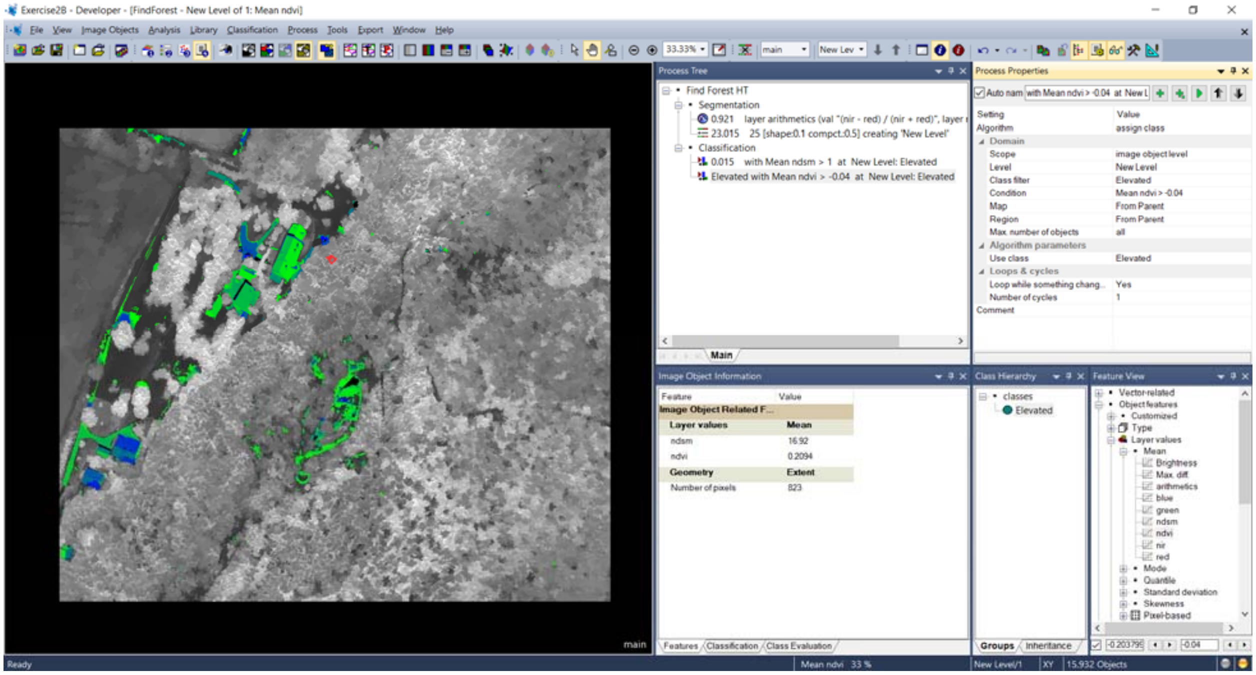
Assign all objects in the image object domain to the class specified by the Use class parameter.

Algorithm parameters

Parameter	Value
Use class	Elevated

Use class
Select class for assignment.

- Here a filter on "Elevated" is being set because "veg" is nested within elevated. So, under the Process Properties, under "Class Filter", choose "Elevated":
- But under "Algorithm Parameters" you want to specify your new class via "use class" = Forest. This is a short cut to creating new classes in your Class Hierarchy (it will be created for you).



Exercise2B - Developer - [FindForest - New Level of 1: Mean ndvi]

Process Tree

- Find Forest HT
  - Segmentation
    - 0.921 layer arithmetics (val "(nir - red) / (nir + red)", layer
    - 23.015 25 [shape:0.1 compct.:0.5] creating 'New Level'
  - Classification
    - 0.015 with Mean ndsm > 1 at New Level: Elevated
    - Elevated with Mean ndvi > -0.04 at New Level: Elevated

Process Properties

☑ Auto nam: with Mean ndvi > -0.04 at New L

Setting	Value
Algorithm	assign class
Domain	
Scope	image object level
Level	New Level
Class filter	Elevated
Condition	Mean ndvi > -0.04
Map	From Parent
Region	From Parent
Max. number of objects	all
Algorithm parameters	
Use class	Elevated
Loops & cycles	
Loop while something chang...	Yes
Number of cycles	1
Comment	

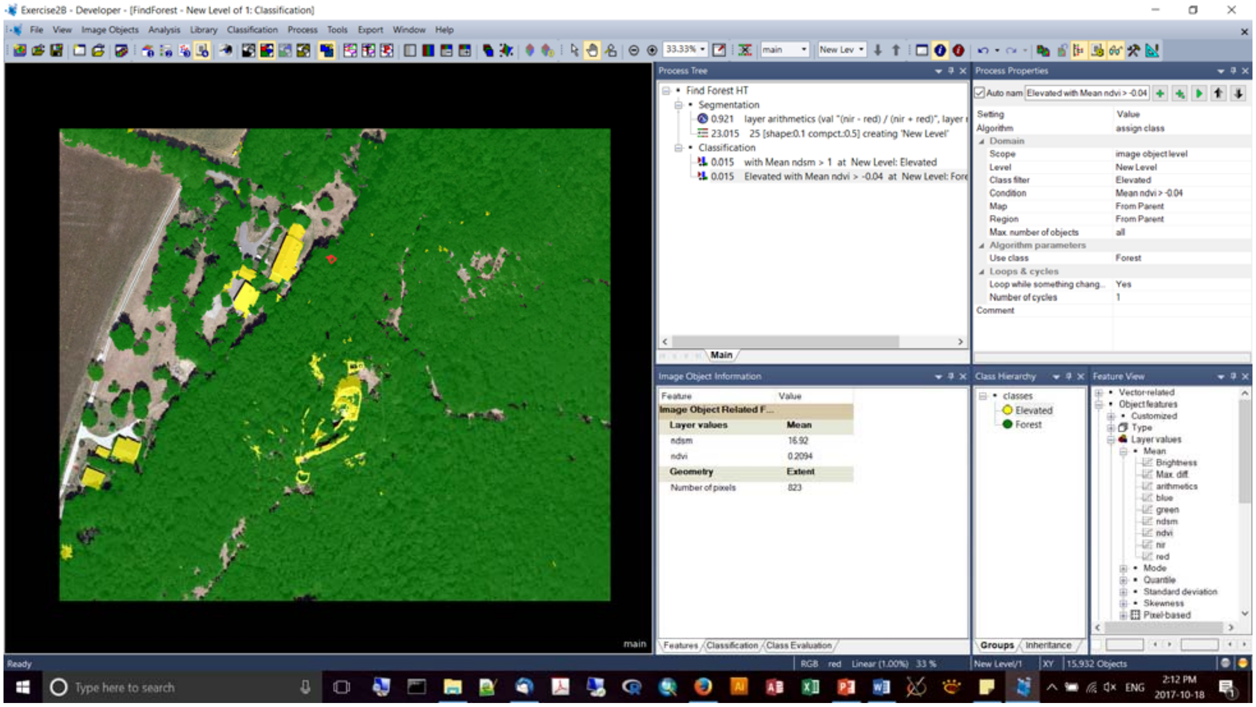
Image Object Information

Feature	Value
Image Object Related F...	
Layer values	Mean
ndsm	16.92
ndvi	0.2094
Geometry	Extent
Number of pixels	823

Class Hierarchy

- classes
  - Elevated

Ready | Mean ndvi 33 % | New Level/1 | XY | 15,932 Objects



Exercise2B - Developer - [FindForest - New Level of 1: Classification]

Process Tree

- Find Forest HT
  - Segmentation
    - 0.921 layer arithmetics (val "(nir - red) / (nir + red)", layer
    - 23.015 25 [shape:0.1 compct.:0.5] creating 'New Level'
  - Classification
    - 0.015 with Mean ndsm > 1 at New Level: Elevated
    - 0.015 Elevated with Mean ndvi > -0.04 at New Level: Fore

Process Properties

☑ Auto nam: Elevated with Mean ndvi > -0.04

Setting	Value
Algorithm	assign class
Domain	
Scope	image object level
Level	New Level
Class filter	Elevated
Condition	Mean ndvi > -0.04
Map	From Parent
Region	From Parent
Max. number of objects	all
Algorithm parameters	
Use class	Forest
Loops & cycles	
Loop while something chang...	Yes
Number of cycles	1
Comment	

Image Object Information

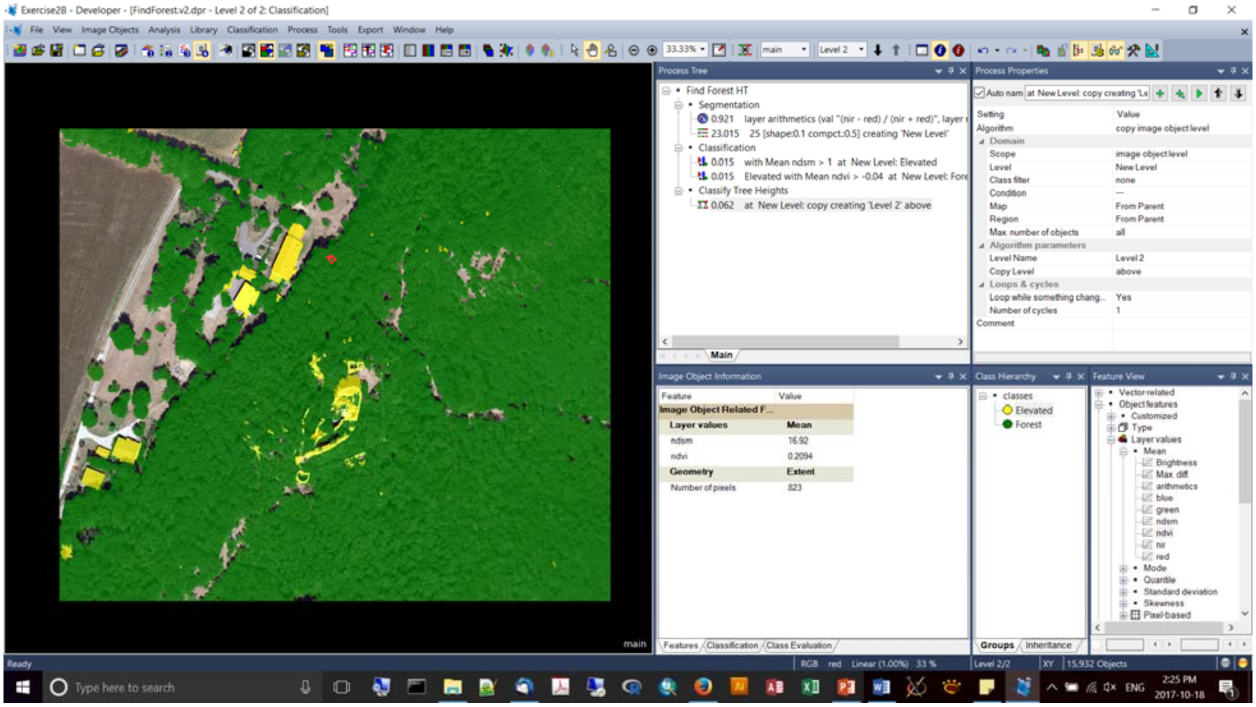
Feature	Value
Image Object Related F...	
Layer values	Mean
ndsm	16.92
ndvi	0.2094
Geometry	Extent
Number of pixels	823

Class Hierarchy

- classes
  - Elevated
  - Forest

Ready | RGB red Linear (1.00%) 33 % | New Level/1 | XY | 15,932 Objects

- Keith then proceeded with a graduated classification of Forest: (1) <5m, (2) 5-10m, (3) 10-15m, (4) 15-20m, (5) > 20m.
- As a note: you can filter out the classes for display (probably not good practice).
- He applied the process “Copy Image object level”


Exercise2B - Developer - [FindForest.v2.dpr - Level 2 of 2: Classification]
File View Image Objects Analysis Library Classification Process Tools Export Window Help
33.33%
main
Level 2
main
Features Classification Class Evaluation
Groups Inheritance
Ready
RGB red Linear (1.00%) 33 %
Level 2/2
XY
15,932 Objects
Type here to search
ENG
2:25 PM
2017-10-18

Process Tree

```
Find Forest HT
  Segmentation
    0.921  layer arithmetics (val "(nir - red) / (nir + red)", layer
    23.015  25 [shape:0.1 compct.:0.5] creating 'New Level'
  Classification
    0.015  with Mean ndsm > 1  at  New Level: Elevated
    0.015  Elevated with Mean ndvi > -0.04  at  New Level: Fore
  Classify Tree Heights
    0.062  at  New Level: copy creating 'Level 2' above
```

Process Properties

☑ Auto nam at New Level: copy creating 'Le

Setting	Value
Algorithm	copy image object level
Domain	
Scope	image object level
Level	New Level
Class filter	none
Condition	---
Map	From Parent
Region	From Parent
Max. number of objects	all
Algorithm parameters	
Level Name	Level 2
Copy Level	above
Loops & cycles	
Loop while something chang...	Yes
Number of cycles	1
Comment	

Image Object Information

Feature	Value
Image Object Related F...	
Layer values	Mean
ndsm	16.92
ndvi	0.2094
Geometry	Extent
Number of pixels	823

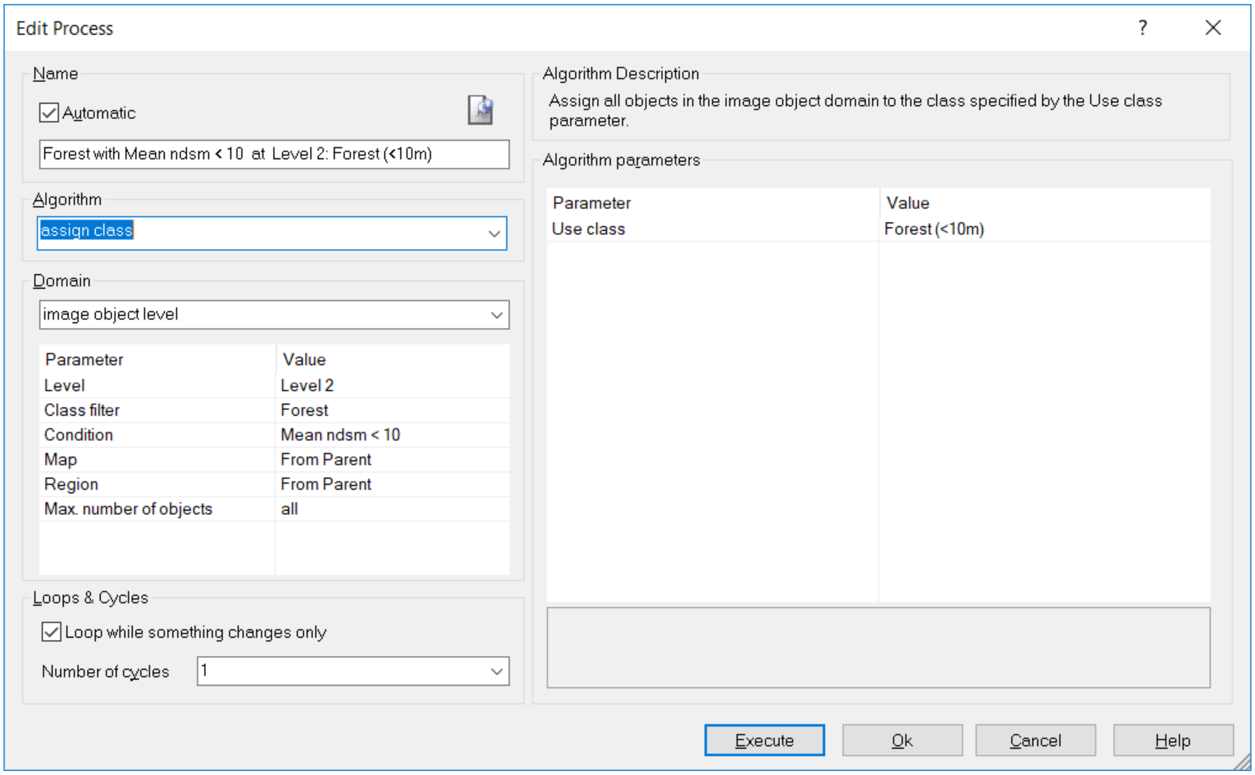
Class Hierarchy

```
classes
  Elevated
  Forest
```

Feature View

```
Vector-related
Object features
  Customized
  Type
  Layer values
    Mean
      Brightness
      Max. diff.
      arithmetics
      blue
      green
      ndsm
      ndvi
      nir
      red
    Mode
    Quantile
    Standard deviation
    Skewness
    Pixel-based
```

- If you screw up your classification at this point, you can use the “X” button to delete the new Level2 and regenerate it from Level 1.
- This is Keith’s logic to copy Level1 to a new level = as eCog has no undo button, this is a good safeguard when you’re working your way through an analysis. If you are unhappy, you can always delete the new Level 2 and recopy it to repair your mangled classification procedure.
- Notice that we will now set the criteria < 10m as there are no longer features <5m in the dataset!



Edit Process

Name

☑ Automatic

Forest with Mean ndsm < 10 at Level 2: Forest (<10m)

Algorithm

assign class

Domain

image object level

Parameter	Value
Level	Level 2
Class filter	Forest
Condition	Mean ndsm < 10
Map	From Parent
Region	From Parent
Max. number of objects	all

Loops & Cycles

☑ Loop while something changes only

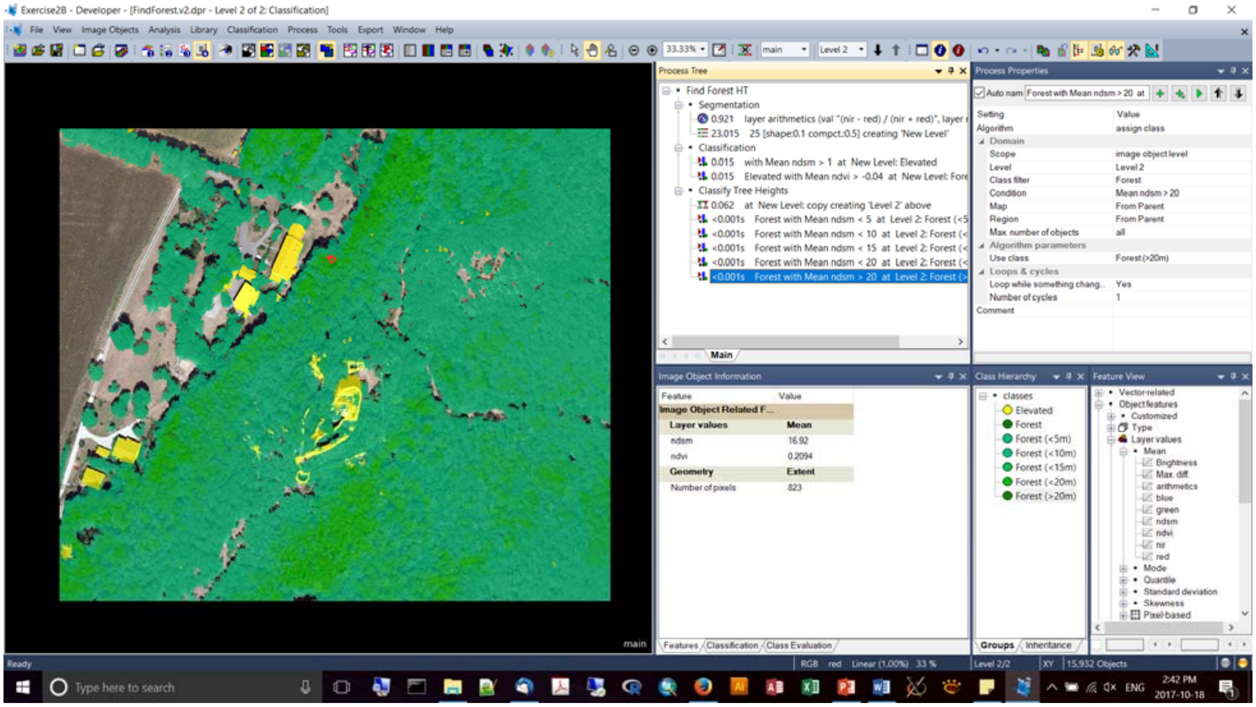
Number of cycles: 1

Algorithm Description

Assign all objects in the image object domain to the class specified by the Use class parameter.

Algorithm parameters

Parameter	Value
Use class	Forest (<10m)



Exercise2B - Developer - [FindForest.v2.dpr - Level 2 of 2: Classification]

Process Tree

```
Find Forest HT
  Segmentation
    0.921   layer arithmetics (val "(nir - red) / (nir + red)", layer
    23.015   25 [shape:0.1 compct.:0.5] creating 'New Level'
  Classification
    0.015   with Mean ndsm > 1  at  New Level: Elevated
    0.015   Elevated with Mean ndvi > -0.04  at  New Level: Fore
  Classify Tree Heights
    0.062   at  New Level: copy creating 'Level 2' above
    <0.001s   Forest with Mean ndsm < 5  at  Level 2: Forest (<5
    <0.001s   Forest with Mean ndsm < 10  at  Level 2: Forest (<
    <0.001s   Forest with Mean ndsm < 15  at  Level 2: Forest (<
    <0.001s   Forest with Mean ndsm < 20  at  Level 2: Forest (<
    <0.001s   Forest with Mean ndsm > 20  at  Level 2: Forest (>
```

Process Properties

☑ Auto nam: Forest with Mean ndsm > 20 at

Setting	Value
Algorithm	assign class
Domain	
Scope	image object level
Level	Level 2
Class filter	Forest
Condition	Mean ndsm > 20
Map	From Parent
Region	From Parent
Max. number of objects	all
Algorithm parameters	
Use class	Forest (>20m)
Loops & cycles	
Loop while something chang...	Yes
Number of cycles	1
Comment	

Image Object Information

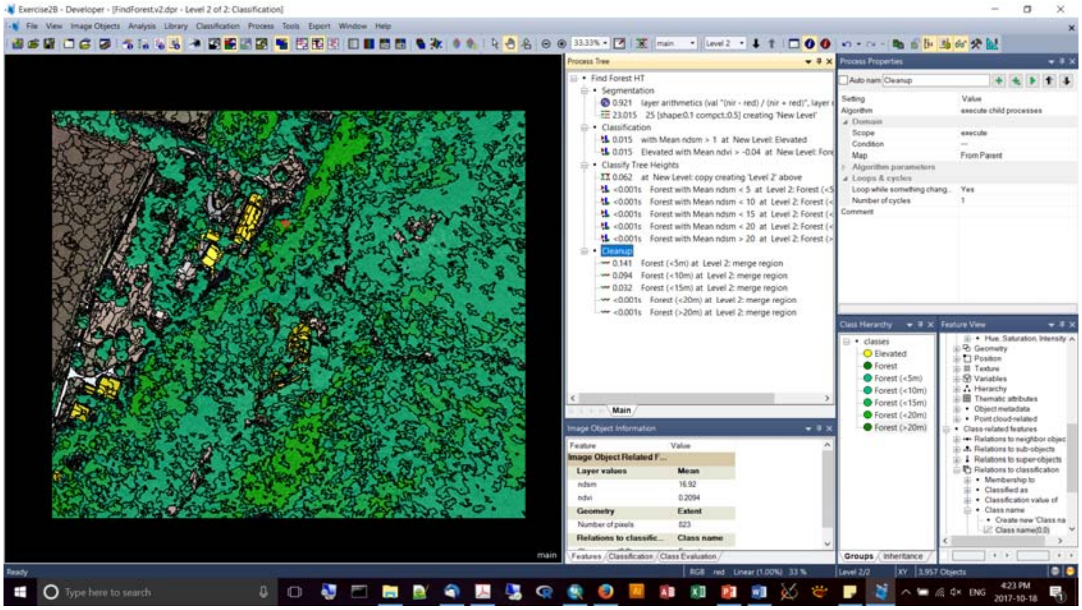
Feature	Value
Image Object Related F...	
Layer values	Mean
ndsm	16.92
ndvi	0.2094
Geometry	Extent
Number of pixels	823

Class Hierarchy

- classes
  - Elevated
  - Forest
  - Forest (<5m)
  - Forest (<10m)
  - Forest (<15m)
  - Forest (<20m)
  - Forest (>20m)

Feature View

- Vector-related
- Object features
  - Customized
  - Type
  - Layer values
    - Mean
      - Brightness
      - Max. diff.
      - arithmetics
      - blue
      - green
      - ndsm
      - ndvi
      - nir
      - red
    - Mode
    - Quantile
    - Standard deviation
    - Skewness
  - Pixel-based

- 

Exercise2B - Developer - [FindForest.v2.dpr - Level 2 of 2: Classification]

Process Tree

```
Find Forest HT
  Segmentation
    0.921    layer arithmetics (val "(nir - red) / (nir + red)", layer
    23.015   25 [shape:0.1 compct.:0.5] creating 'New Level'
  Classification
    0.015    with Mean ndsm > 1  at  New Level: Elevated
    0.015    Elevated with Mean ndvi > -0.04  at  New Level: Fore
  Classify Tree Heights
    0.062    at  New Level: copy creating 'Level 2' above
    <0.001s  Forest with Mean ndsm < 5  at  Level 2: Forest (<5
    <0.001s  Forest with Mean ndsm < 10  at  Level 2: Forest (<
    <0.001s  Forest with Mean ndsm < 15  at  Level 2: Forest (<
    <0.001s  Forest with Mean ndsm < 20  at  Level 2: Forest (<
    <0.001s  Forest with Mean ndsm > 20  at  Level 2: Forest (>
  Cleanup
    0.141    Forest (<5m) at  Level 2: merge region
    0.094    Forest (<10m) at  Level 2: merge region
    0.032    Forest (<15m) at  Level 2: merge region
    <0.001s  Forest (<20m) at  Level 2: merge region
    <0.001s  Forest (>20m) at  Level 2: merge region
```

Process Properties

Auto nam: Cleanup

Setting	Value
Algorithm	execute child processes
Domain	
Scope	execute
Condition	—
Map	From Parent
Algorithm parameters	
Loops & cycles	
Loop while something chang...	Yes
Number of cycles	1
Comment	

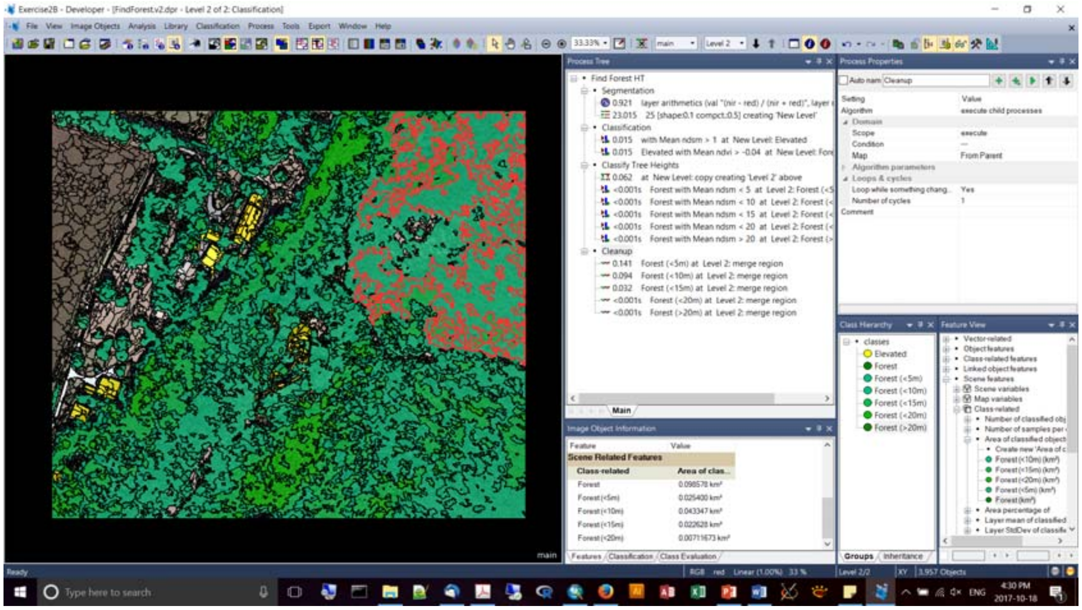
Image Object Information

Feature	Value
Image Object Related F...	
Layer values	Mean
ndsm	16.92
ndvi	0.2094
Geometry	Extent
Number of pixels	823
Relations to classific...	Class name

Class Hierarchy

- classes
  - Elevated
  - Forest
  - Forest (<5m)
  - Forest (<10m)
  - Forest (<15m)
  - Forest (<20m)
  - Forest (>20m)

- You can retrieve “Feature View” statistics on the classified objects by selecting “Scene Features”, Class Related, Area of Classified Objects.



Exercise2B - Developer - [FindForest.v2.dpr - Level 2 of 2: Classification]

Image Object Information

Feature	Value
Scene Related Features	
Class-related	Area of clas...
Forest	0.098578 km²
Forest (<5m)	0.025400 km²
Forest (<10m)	0.043347 km²
Forest (<15m)	0.022628 km²
Forest (<20m)	0.00711673 km²

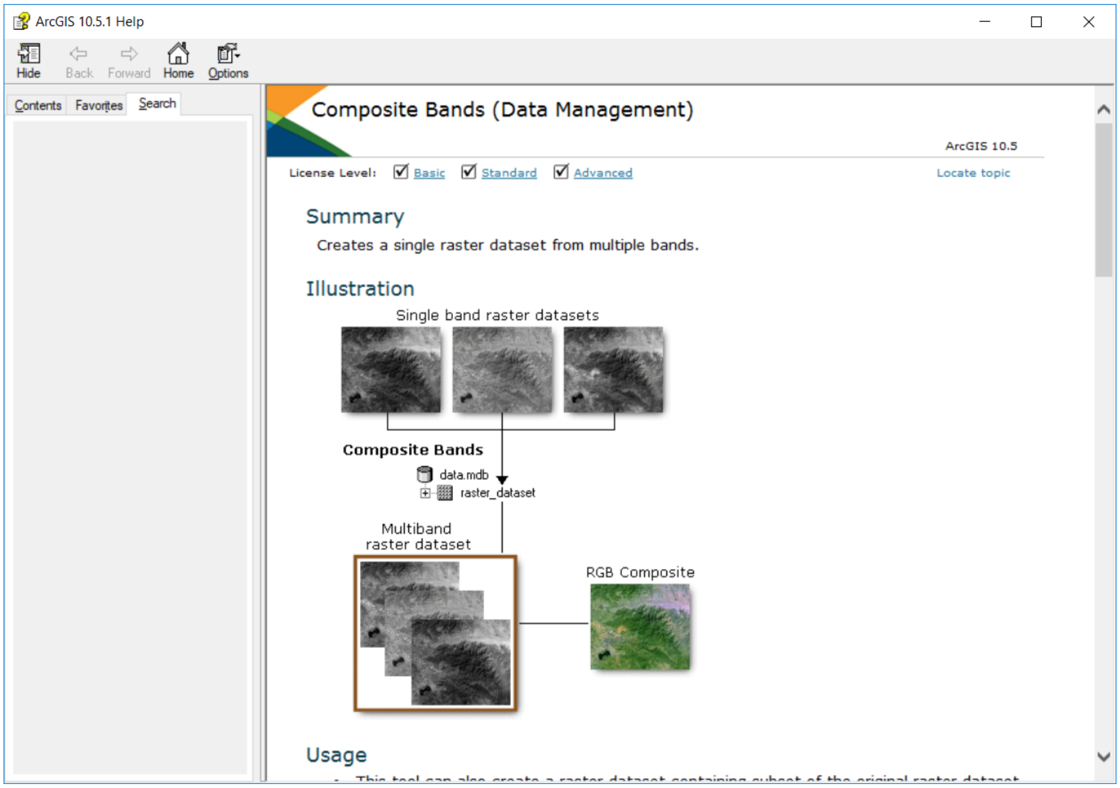
Feature View

- Vector-related
- Object features
- Class-related features
- Linked object features
- Scene features
  - Scene variables
  - Map variables
  - Class-related
    - Number of classified obj
    - Number of samples per
    - Area of classified object
      - Create new 'Area of c
      - Forest (<10m) (km²)
      - Forest (<15m) (km²)
      - Forest (<20m) (km²)
      - Forest (<5m) (km²)
      - Forest (km²)
    - Area percentage of
    - Layer mean of classified
    - Layer StdDev of classifie

File View Image Objects Analysis Library Classification Process Tools Export Window Help
Process Tree
Process Properties
Class Hierarchy
Main
Features Classification Class Evaluation
Groups Inheritance
Ready
RGB red Linear (1.00%) 33 %
Level 2/2
XY
3,957 Objects
Type here to search
ENG
4:30 PM
2017-10-18

Exercise #3

- Supervised classification – just need to select samples for the algorithm, which classifies for you.
- We'll look at an area north of Denver to identify agricultural areas.
- We'll combine multiple segmentations.
- We'll apply supervised classification, then do some manual editing, followed by refinement and export.
- Data from ESA Sentinel-2 (B,G,R,NIR). Sentinel-2 doesn't have thermal bands, but it has three different bands in the red-edge which is very good for vegetation classification.
- Multiple band combination via ESRI = Composite Bands (inside the Raster tools).



Composite Bands (Data Management)

ArcGIS 10.5

License Level: ☑ Basic ☑ Standard ☑ Advanced   Locate topic

Summary

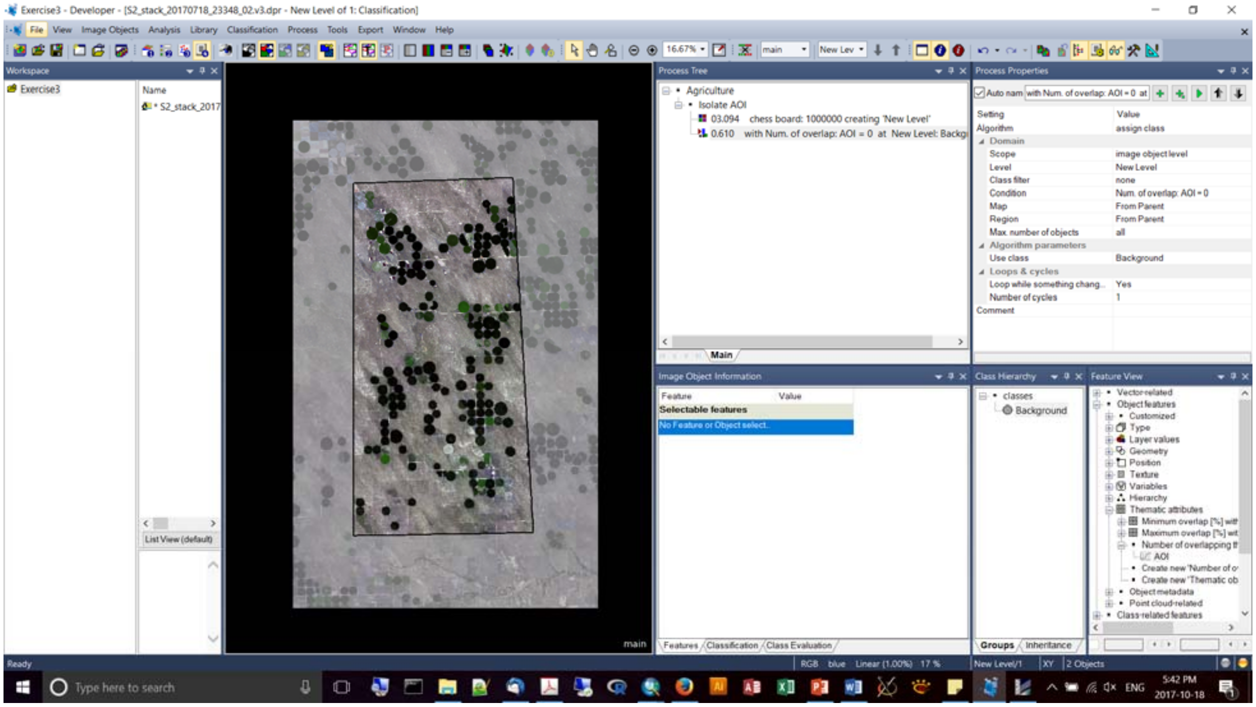
Creates a single raster dataset from multiple bands.

Illustration

Usage

- This tool can also create a raster dataset containing subset of the original raster dataset

- Use the chessboard algorithm with a very large pixel size (1,000,000) and an AOI to segment your study area into inside and outside the AOI.
- Access the "Num of Overlap" as a Feature property "Create new 'Number of overlapping thematic objects". This will set a binary value of 1 or 0 depending upon whether a pixel is inside or outside the AOI (or other thematic boundary layer).
- Set an "Assign Class" algorithm, choosing a Condition of "Num. of overlap -= 0", defining it as "Background". Apparently its easier to work with this way as you can focus your subsequent ruleset on portions of the image NOT background.


Exercise3 - Developer - [S2_stack_20170718_23348_02.v3.dpr - New Level of 1: Classification]
File View Image Objects Analysis Library Classification Process Tools Export Window Help
Workspace
Exercise3
Name
S2_stack_2017
Image Object Information
Class Hierarchy
classes
Background
Feature View
Ready

Process Tree

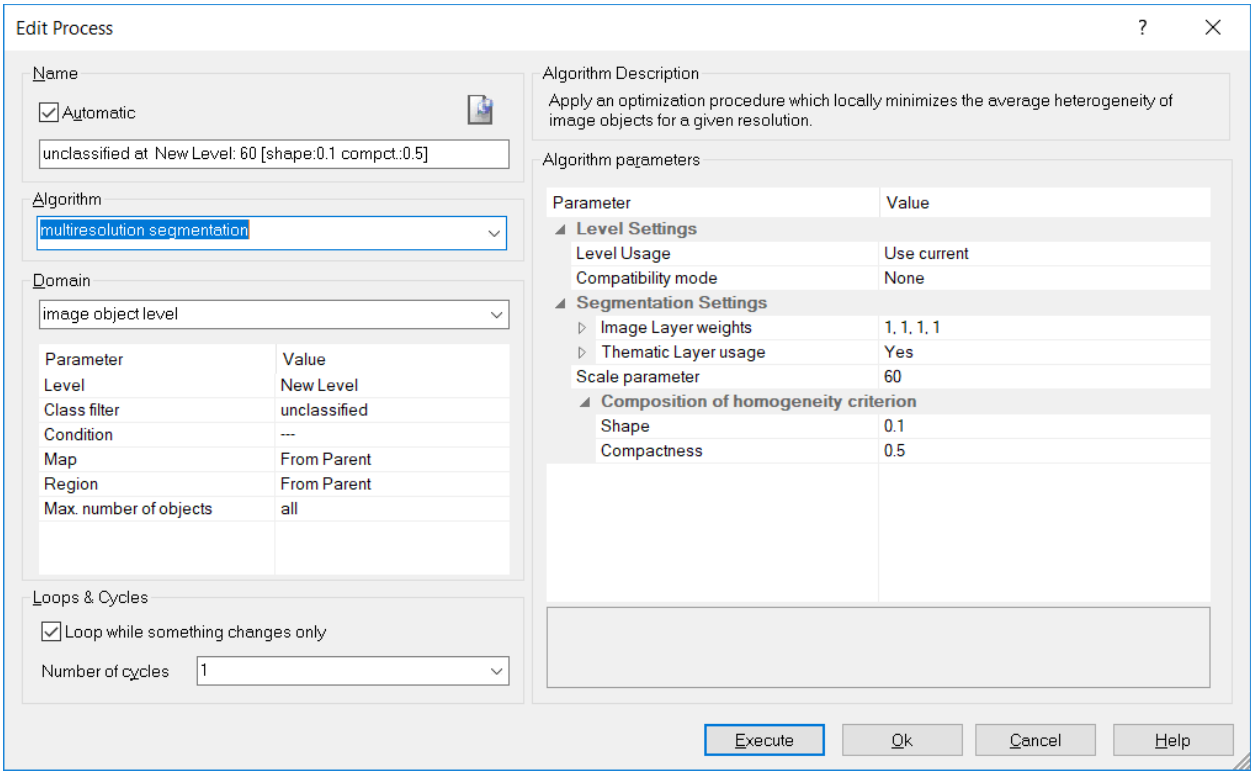
- Agriculture
  - Isolate AOI
    - 03.094 chess board: 1000000 creating 'New Level'
    - 0.610 with Num. of overlap: AOI = 0 at New Level: Backgi

Process Properties

☑ Auto nam with Num. of overlap: AOI = 0 at

Setting	Value
Algorithm	assign class
Domain	
Scope	image object level
Level	New Level
Class filter	none
Condition	Num. of overlap: AOI = 0
Map	From Parent
Region	From Parent
Max. number of objects	all
Algorithm parameters	
Use class	Background
Loops & cycles	
Loop while something chang...	Yes
Number of cycles	1
Comment	

- Then, when you do your segmentation on the area of interest, you can set your Class Filter to be "unclassified" – this will focus the Multiresolution segmentation only on the portions of the image NOT in the "Background", i.e., inside your area of interest~
- Set to Level Usage to "Use Current".
- Scale parameter set to 60.



Edit Process

Name

☑ Automatic

unclassified at New Level: 60 [shape:0.1 compct.:0.5]

Algorithm

multiresolution segmentation

Domain

image object level

Parameter	Value
Level	New Level
Class filter	unclassified
Condition	---
Map	From Parent
Region	From Parent
Max. number of objects	all

Loops & Cycles

☑ Loop while something changes only

Number of cycles: 1

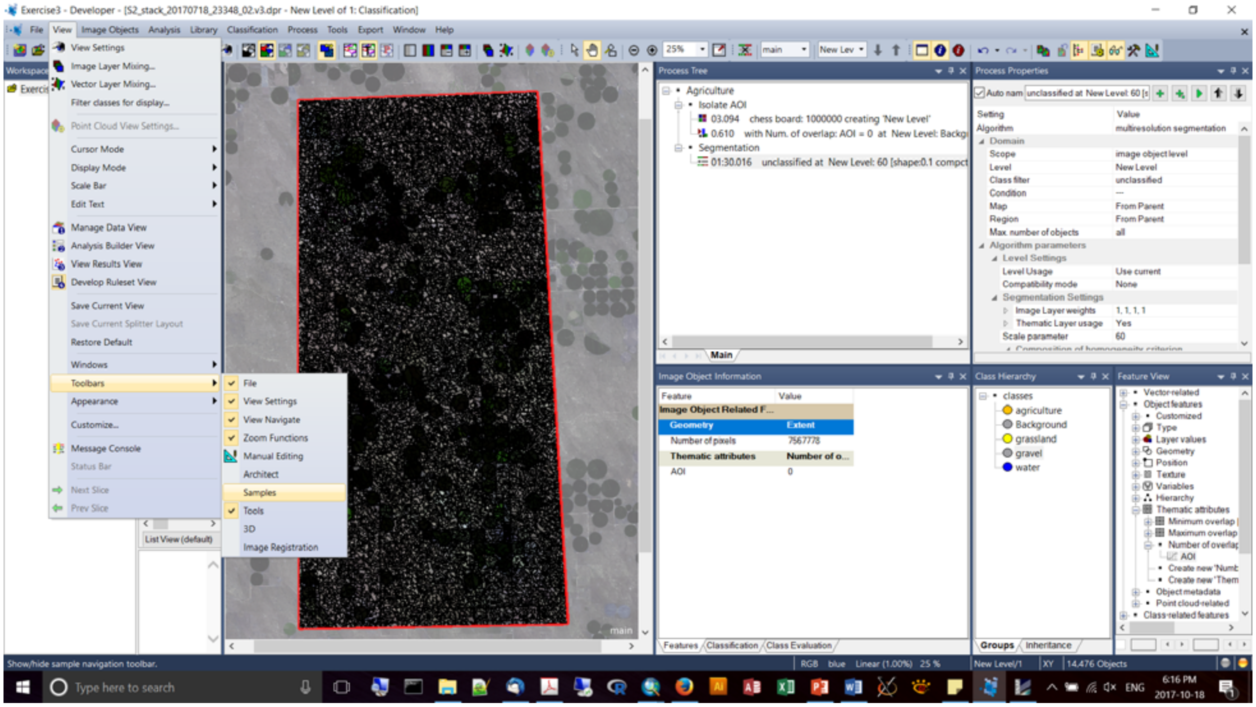
Algorithm Description

Apply an optimization procedure which locally minimizes the average heterogeneity of image objects for a given resolution.

Algorithm parameters

Parameter	Value
Level Settings	
Level Usage	Use current
Compatibility mode	None
Segmentation Settings	
Image Layer weights	1, 1, 1, 1
Thematic Layer usage	Yes
Scale parameter	60
Composition of homogeneity criterion	
Shape	0.1
Compactness	0.5

- Accessing the sampling tools;



Exercise3 - Developer - [S2_stack_20170718_23348_02.v3.dpr - New Level of 1: Classification]

Process Tree

```
Agriculture
  Isolate AOI
    03.094   chess board: 1000000 creating 'New Level'
    0.610   with Num. of overlap: AOI = 0  at  New Level: Backgr
  Segmentation
    01:30.016   unclassified at  New Level: 60 [shape:0.1 compct
```

Process Properties

☑ Auto nam unclassified at New Level: 60 [s

Setting	Value
Algorithm	multiresolution segmentation
Domain	
Scope	image object level
Level	New Level
Class filter	unclassified
Condition	---
Map	From Parent
Region	From Parent
Max. number of objects	all
Algorithm parameters	
Level Settings	
Level Usage	Use current
Compatibility mode	None
Segmentation Settings	
Image Layer weights	1, 1, 1, 1
Thematic Layer usage	Yes
Scale parameter	60
Composition of homogeneity criterion	

Image Object Information

Feature	Value
Image Object Related F...	
Geometry	Extent
Number of pixels	7567778
Thematic attributes	Number of o...
AOI	0

Class Hierarchy

```
classes
  agriculture
  Background
  grassland
  gravel
  water
```

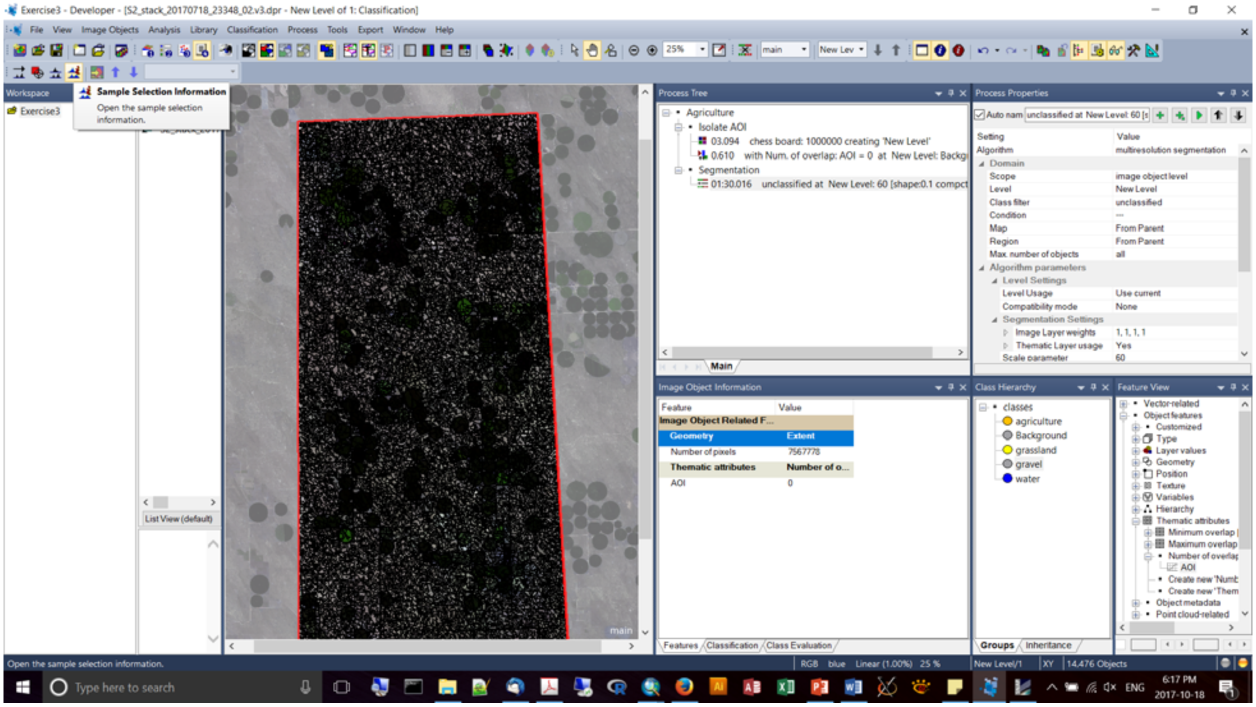
Feature View

```
Vector-related
Object features
  Customized
  Type
  Layer values
  Geometry
  Position
  Texture
  Variables
  Hierarchy
  Thematic attributes
    Minimum overlap |
    Maximum overlap
    Number of overlap
      AOI
    Create new 'Numb
    Create new 'Them
  Object metadata
  Point cloud-related
Class-related features
```

Show/hide sample navigation toolbar.

RGB blue Linear (1.00%) 25 % New Level/1 XY 14,476 Objects

Type here to search ENG 6:16 PM 2017-10-18



Sample Selection Information

Open the sample selection information.

```
Agriculture
  Isolate AOI
    03.094   chess board: 1000000 creating 'New Level'
    0.610    with Num. of overlap: AOI = 0  at  New Level: Backg
  Segmentation
    01:30.016   unclassified at  New Level: 60 [shape:0.1 compct
```

☑ Auto nam unclassified at New Level: 60 [s

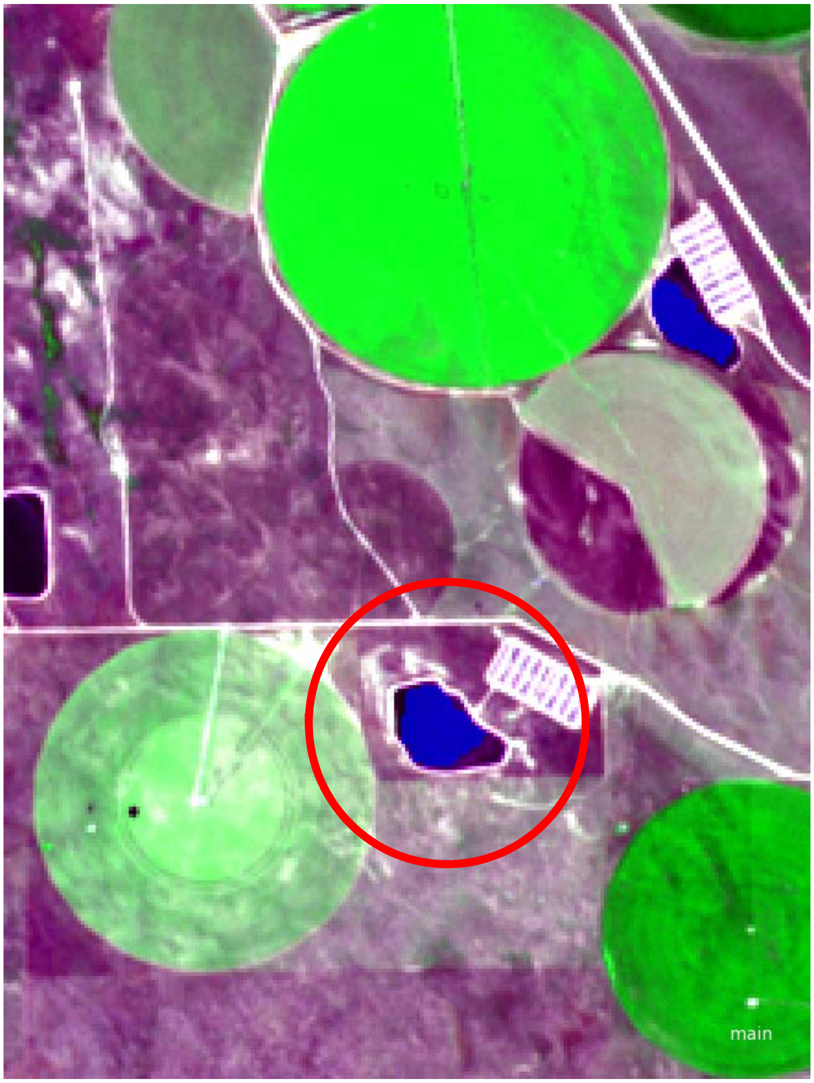
Setting	Value
Algorithm	multiresolution segmentation
Domain	
Scope	image object level
Level	New Level
Class filter	unclassified
Condition	---
Map	From Parent
Region	From Parent
Max. number of objects	all
Algorithm parameters	
Level Settings	
Level Usage	Use current
Compatibility mode	None
Segmentation Settings	
Image Layer weights	1, 1, 1, 1
Thematic Layer usage	Yes
Scale parameter	60

Feature	Value
Image Object Related F...	
Geometry	Extent
Number of pixels	7567778
Thematic attributes	Number of o...
AOI	0

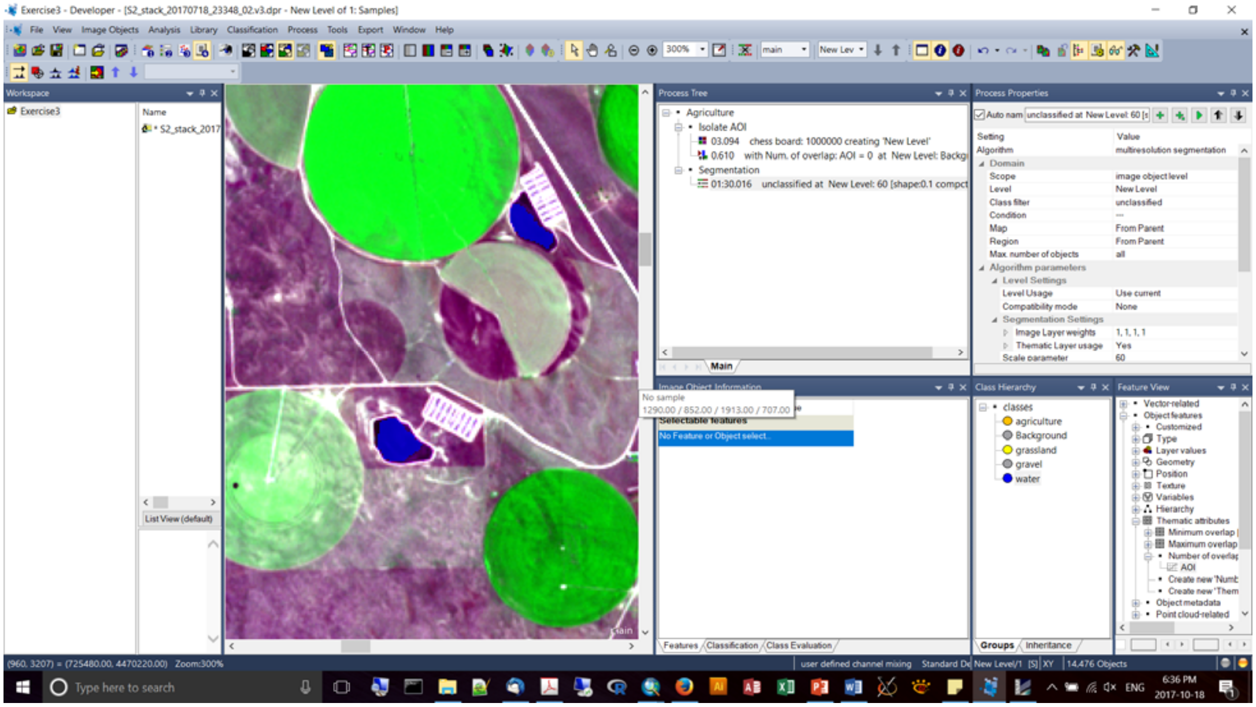
Open the sample selection information.

RGB blue Linear (1.00%) 25 % | New Level/1 | XY | 14,476 Objects

- Gathering sample: click on the class in your hierarchy (e.g., “agriculture”), activate the “Select Samples”, and double click on features in the image. Every time you do this you will add objects to the “selection” bin.
- Just as a note: if you click on the mouse button, you can drag your surface while still in edit mode.



- When you've finished selecting, add a "Classification" chapter to your tree, and create a new **classifier**, in this case choose a support vector machine, and choose to use only your selection. This step will create the model (e.g., "Model_01").
- Execute, then create a second classifier process, but this time apply a filter to the "unclassified" objects in the image. This will apply the model predictions. For "operation" it will be "Apply".



Exercise3 - Developer - [S2_stack_20170718_23348_02.v3.dpr - New Level of 1: Samples]

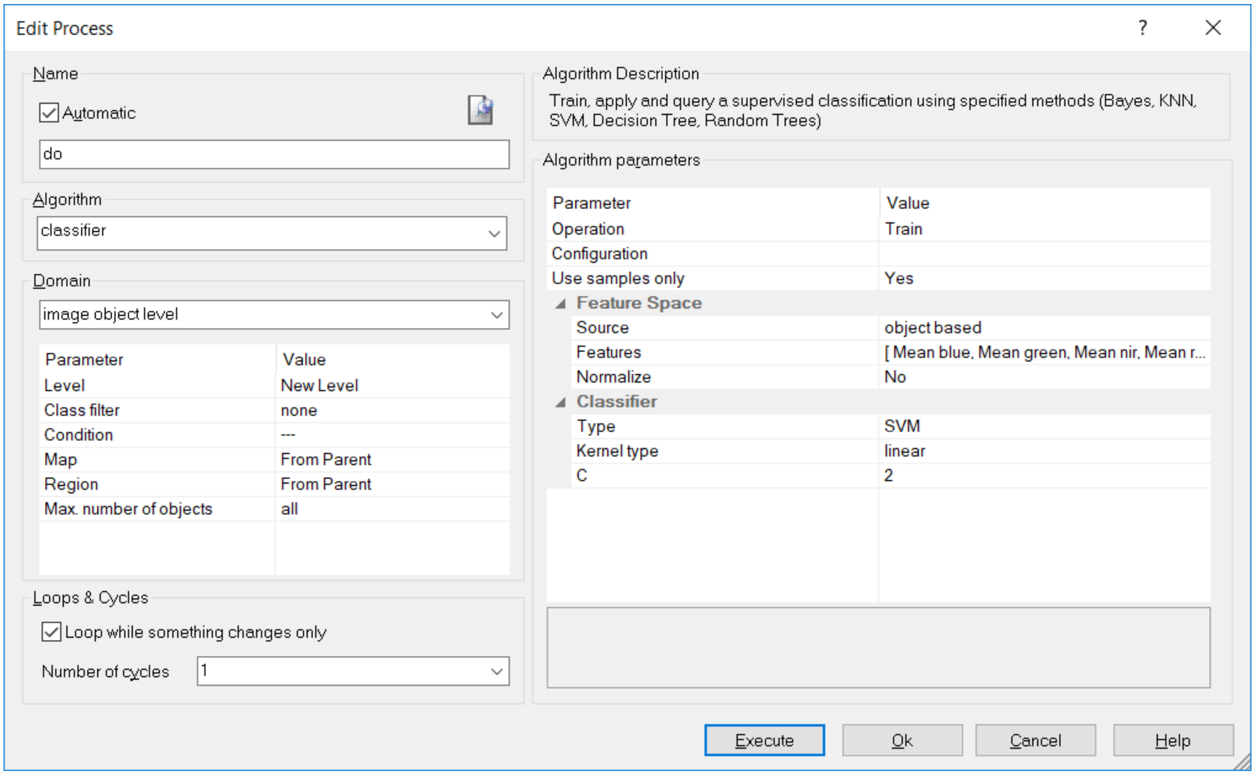
Process Tree

- Agriculture
  - Isolate AOI
    - 03.094 chess board: 1000000 creating 'New Level'
    - 0.610 with Num. of overlap: AOI = 0 at New Level: Backgr
  - Segmentation
    - 01:30.016 unclassified at New Level: 60 [shape:0.1 compct

Process Properties

☑ Auto nam unclassified at New Level: 60 [s

Setting	Value
Algorithm	multiresolution segmentation
Domain	
Scope	image object level
Level	New Level
Class filter	unclassified
Condition	---
Map	From Parent
Region	From Parent
Max. number of objects	all
Algorithm parameters	
Level Settings	
Level Usage	Use current
Compatibility mode	None
Segmentation Settings	
Image Layer weights	1, 1, 1, 1
Thematic Layer usage	Yes
Scale parameter	60



Edit Process

Name

☑ Automatic

do

Algorithm

classifier

Domain

image object level

Parameter	Value
Level	New Level
Class filter	none
Condition	---
Map	From Parent
Region	From Parent
Max. number of objects	all

Loops & Cycles

☑ Loop while something changes only

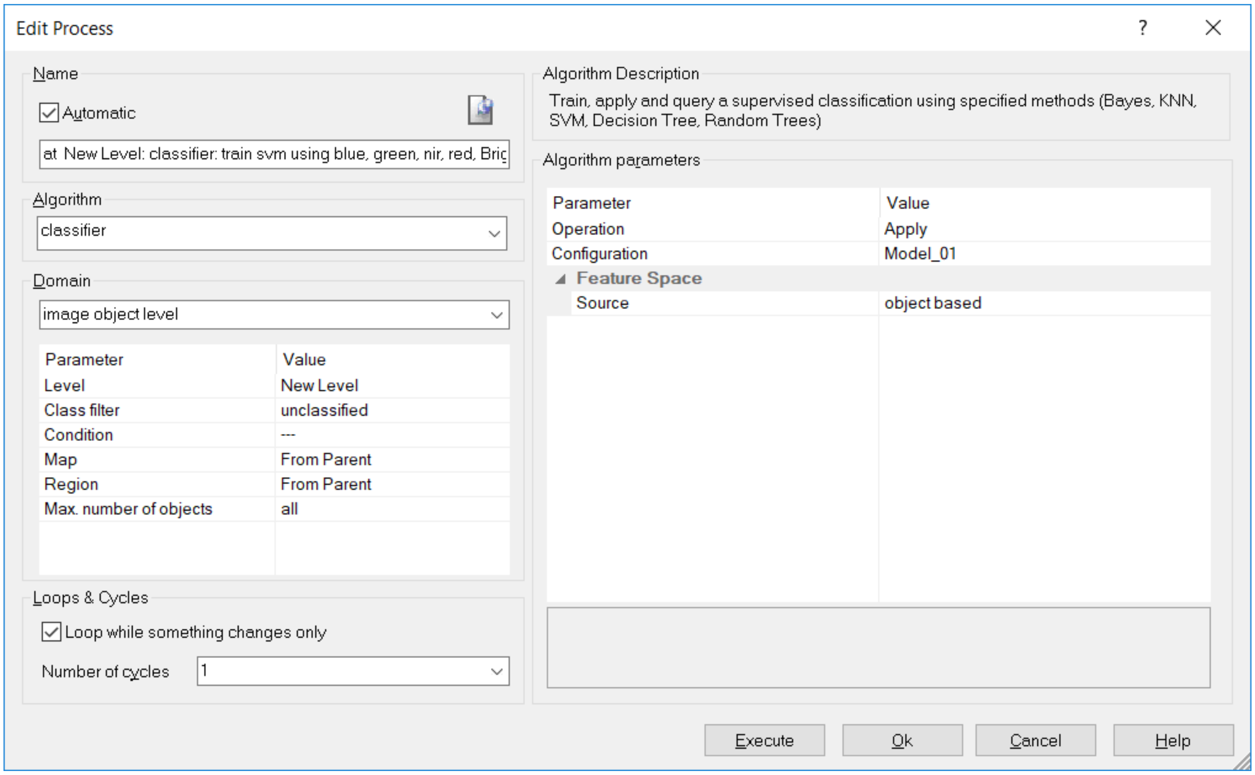
Number of cycles 1

Algorithm Description

Train, apply and query a supervised classification using specified methods (Bayes, KNN, SVM, Decision Tree, Random Trees)

Algorithm parameters

Parameter	Value
Operation	Train
Configuration	
Use samples only	Yes
Feature Space	
Source	object based
Features	[Mean blue, Mean green, Mean nir, Mean r...
Normalize	No
Classifier	
Type	SVM
Kernel type	linear
C	2



Edit Process

Name

☑ Automatic

at New Level: classifier: train svm using blue, green, nir, red, Bric

Algorithm

classifier

Domain

image object level

Parameter	Value
Level	New Level
Class filter	unclassified
Condition	---
Map	From Parent
Region	From Parent
Max. number of objects	all

Loops & Cycles

☑ Loop while something changes only

Number of cycles: 1

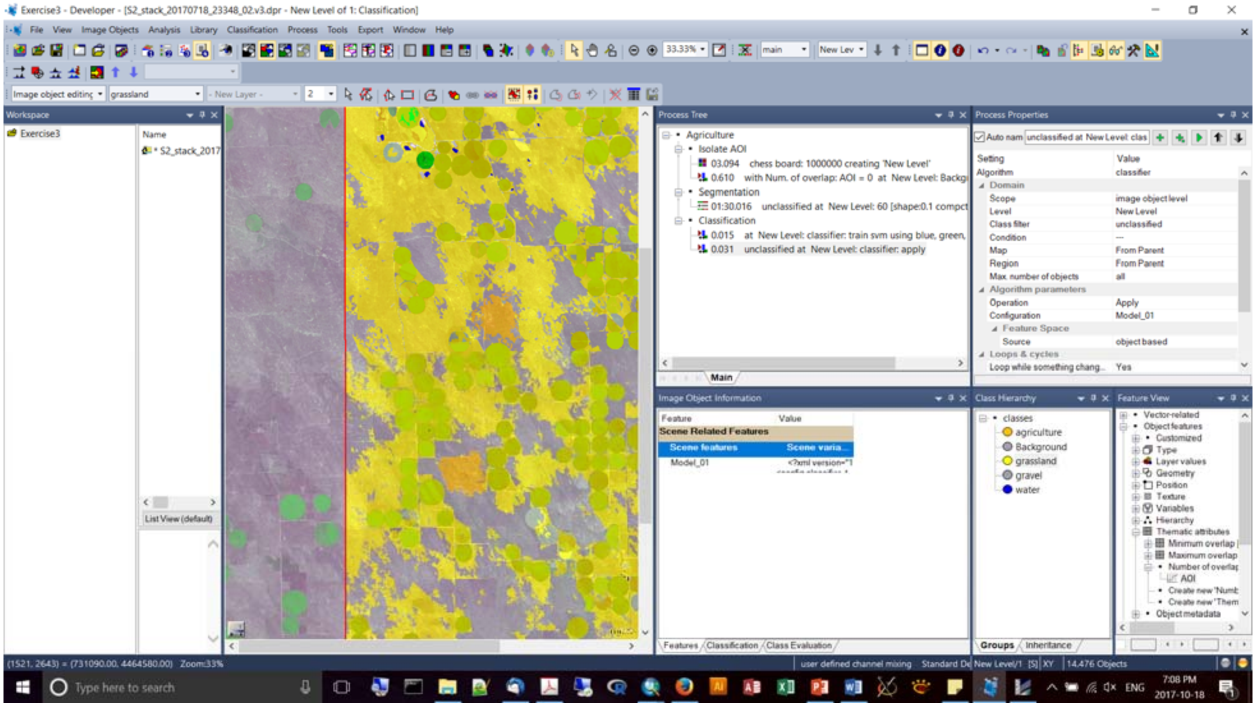
Algorithm Description

Train, apply and query a supervised classification using specified methods (Bayes, KNN, SVM, Decision Tree, Random Trees)

Algorithm parameters

Parameter	Value
Operation	Apply
Configuration	Model_01
Feature Space	
Source	object based

- It is possible to manually edit the surface,
- Under View, Toolbars, choose the “Manual Editing” toolbar.
- Look at the toolbar – you’ll see it’s set on a particular class, e.g., “grassland”, so basically and objects you select on the screen can be assigned to the class grassland.



Exercise3 - Developer - [S2_stack_20170718_23348_02.v3.dpr - New Level of 1: Classification]

Process Tree

```
⊟ • Agriculture
   ⊟ • Isolate AOI
        03.094   chess board: 1000000 creating 'New Level'
        0.610    with Num. of overlap: AOI = 0  at  New Level: Backgi
   ⊟ • Segmentation
        01:30.016   unclassified at  New Level: 60 [shape:0.1 compct
   ⊟ • Classification
        0.015    at  New Level: classifier: train svm using blue, green,
        0.031    unclassified at  New Level: classifier: apply
```

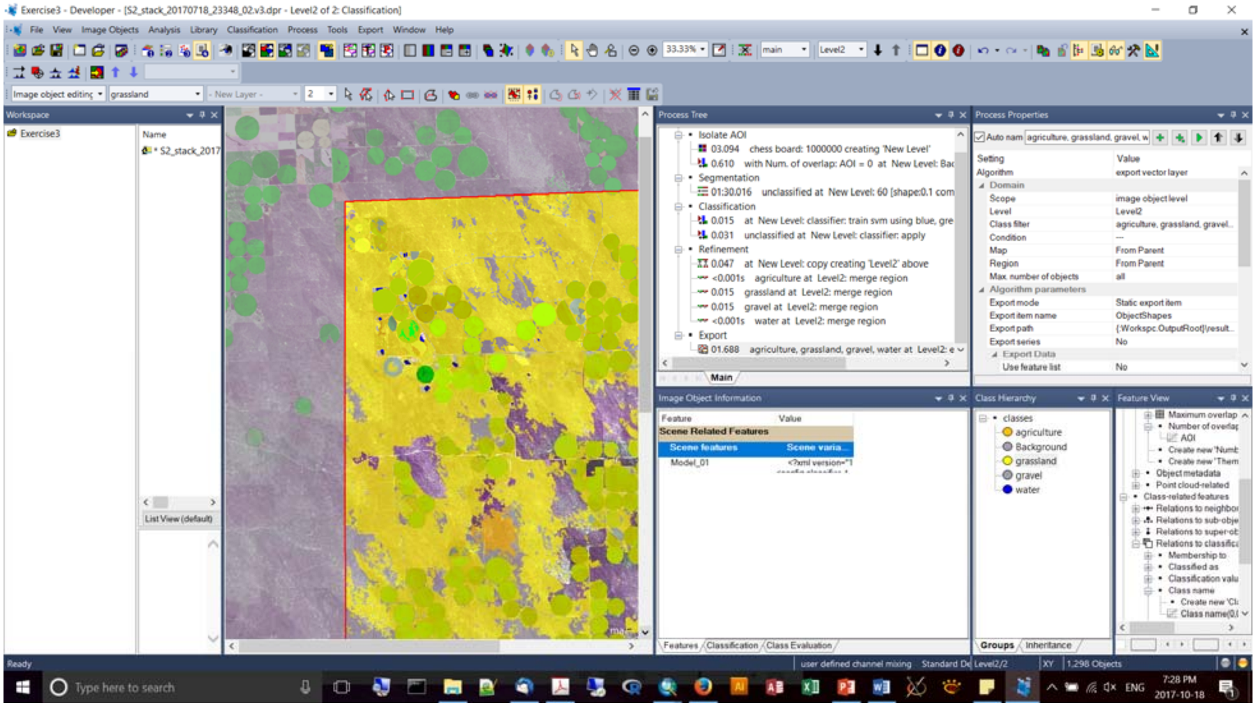
Process Properties

☑ Auto nam unclassified at New Level: clas

Setting	Value
Algorithm	classifier
Domain	
Scope	image object level
Level	New Level
Class filter	unclassified
Condition	---
Map	From Parent
Region	From Parent
Max. number of objects	all
Algorithm parameters	
Operation	Apply
Configuration	Model_01
Feature Space	
Source	object based
Loops & cycles	
Loop while something chang...	Yes

Class Hierarchy: classes — agriculture, Background, grassland, gravel, water

- As we saw before, a good final step will be to Copy your level to a new level (Level 2), and then add some processes for merging the objects prior to export.



Exercise3 - Developer - [S2_stack_20170718_23348_02.v3.dpr - Level2 of 2: Classification]

Process Tree

```
   ⊟ • Isolate AOI
        03.094   chess board: 1000000 creating 'New Level'
        0.610    with Num. of overlap: AOI = 0  at  New Level: Bac
   ⊟ • Segmentation
        01:30.016   unclassified at  New Level: 60 [shape:0.1 com
   ⊟ • Classification
        0.015    at  New Level: classifier: train svm using blue, gre
        0.031    unclassified at  New Level: classifier: apply
   ⊟ • Refinement
        0.047    at  New Level: copy creating 'Level2' above
        <0.001s   agriculture at  Level2: merge region
        0.015    grassland at  Level2: merge region
        0.015    gravel at  Level2: merge region
        <0.001s   water at  Level2: merge region
   ⊟ • Export
        01.688   agriculture, grassland, gravel, water at  Level2: e
```

Process Properties

☑ Auto nam agriculture, grassland, gravel, w

Setting	Value
Algorithm	export vector layer
Domain	
Scope	image object level
Level	Level2
Class filter	agriculture, grassland, gravel...
Condition	---
Map	From Parent
Region	From Parent
Max. number of objects	all
Algorithm parameters	
Export mode	Static export item
Export item name	ObjectShapes
Export path	{:Workspc.OutputRoot}\result...
Export series	No
Export Data	
Use feature list	No

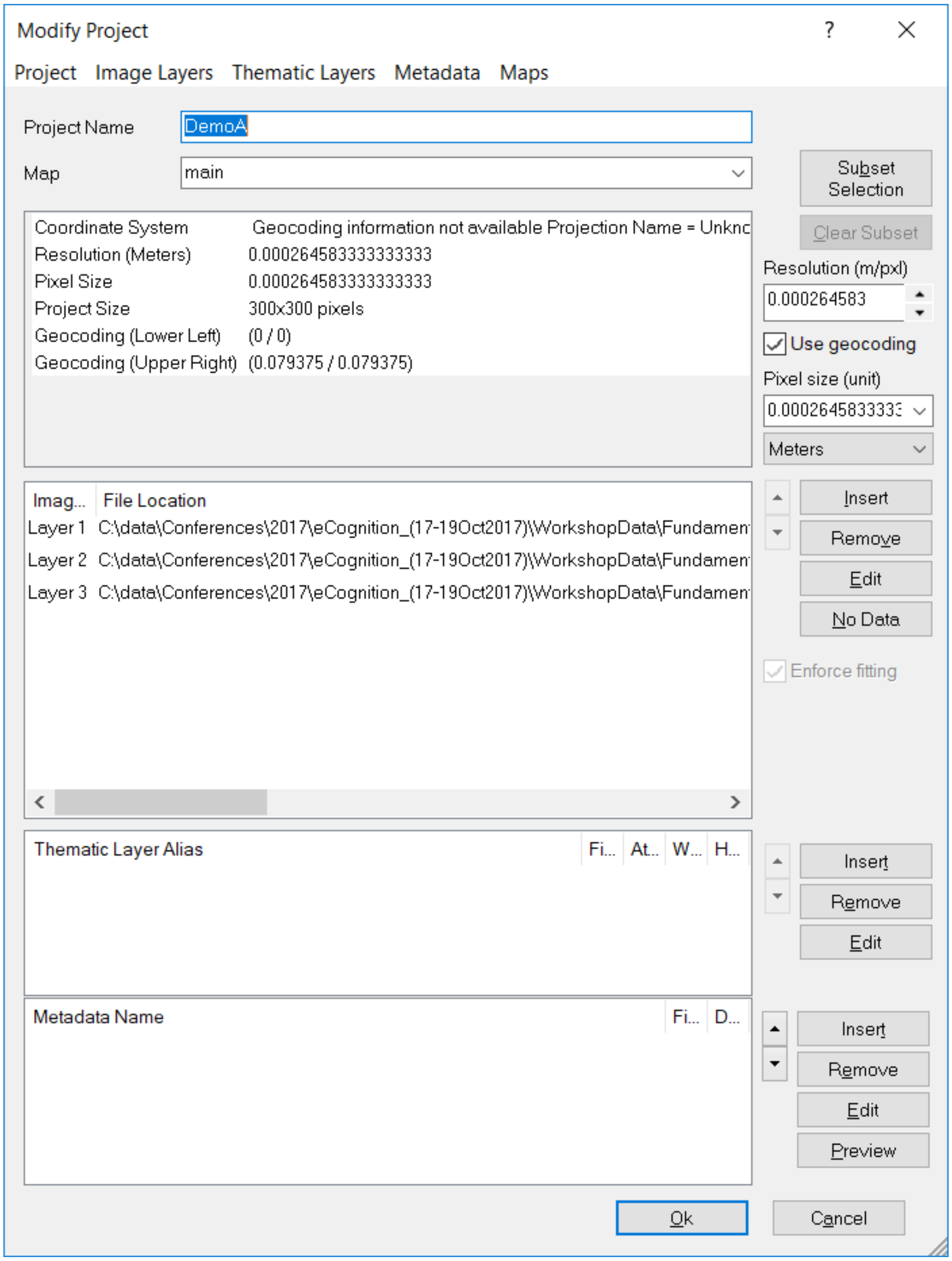
Class Hierarchy: classes — agriculture, Background, grassland, gravel, water

- As we saw before, a good final step will be to Copy your level to a new level (Level 2), and then add some processes for merging the objects prior to export.

- It's possible to apply training locations from other images using a model training approach involving the creation and importation of "Sample Statistics".
- This involves a .csv export of the characteristics of your sample objects, which you can then bring into a training and application in other landscapes / images.

OCT.19/17

- Template matching to identify particular objects (this is new data provided by the instructors last minute).
- New work by eCog on CNN (neural networks) is very promising and generates much lower error rates than with template matching (e.g., 6.8% vs. 22%).
- To load two layers, not in the same space, you can load mulitple maps within single projects.
- Templates for our trees will be created from two different locations.



Project Name: DemoA

Map: main

Coordinate System	Geocoding information not available Projection Name = Unknc
Resolution (Meters)	0.000264583333333333
Pixel Size	0.000264583333333333
Project Size	300x300 pixels
Geocoding (Lower Left)	(0 / 0)
Geocoding (Upper Right)	(0.079375 / 0.079375)

Resolution (m/pxl): 0.000264583

☑ Use geocoding

Pixel size (unit): 0.00026458333ɜ

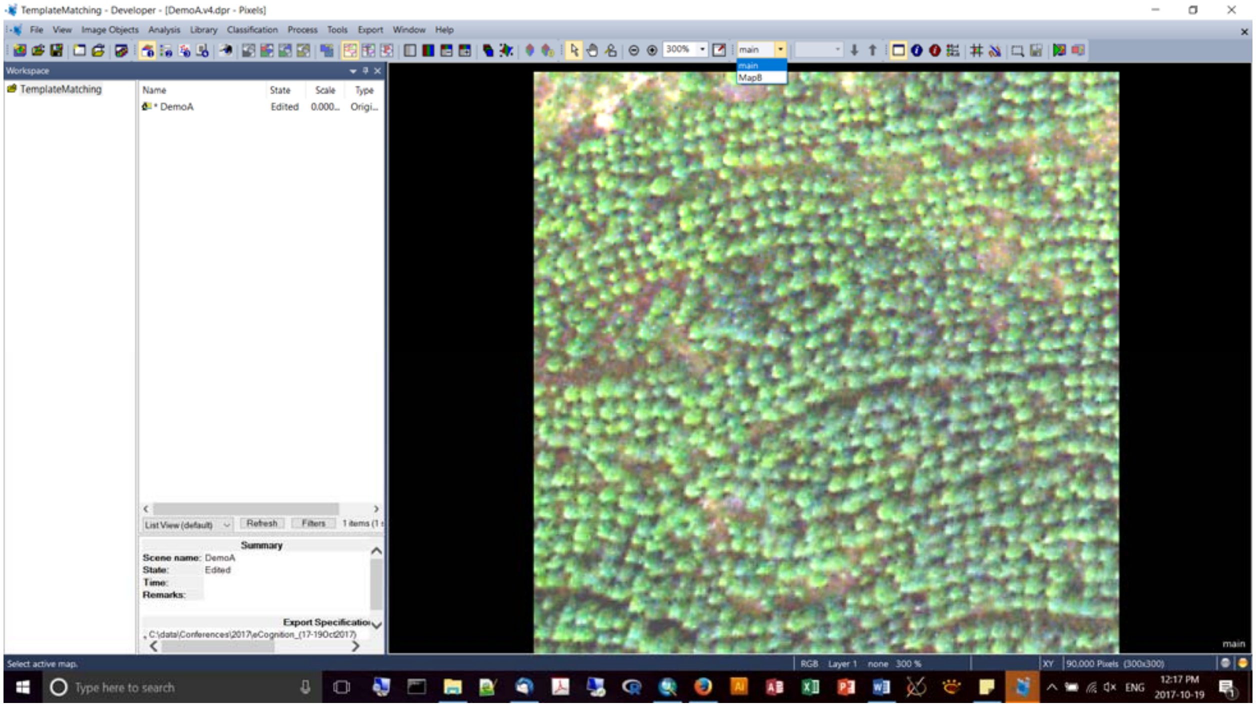
Meters

Imag...	File Location
Layer 1	C:\data\Conferences\2017\eCognition_(17-19Oct2017)\WorkshopData\Fundamen
Layer 2	C:\data\Conferences\2017\eCognition_(17-19Oct2017)\WorkshopData\Fundamen
Layer 3	C:\data\Conferences\2017\eCognition_(17-19Oct2017)\WorkshopData\Fundamen

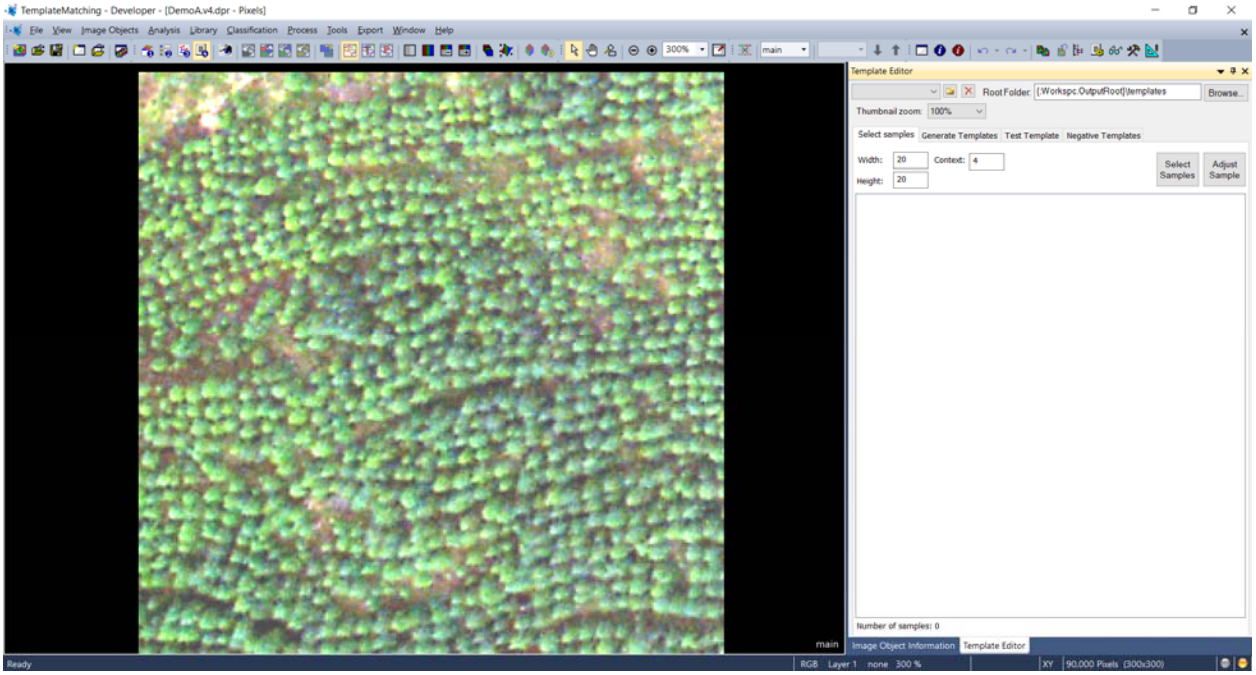
Thematic Layer Alias	Fi...	At...	W...	H...

Metadata Name	Fi...	D...

- Notice how you can toggle between maps in the main menu:

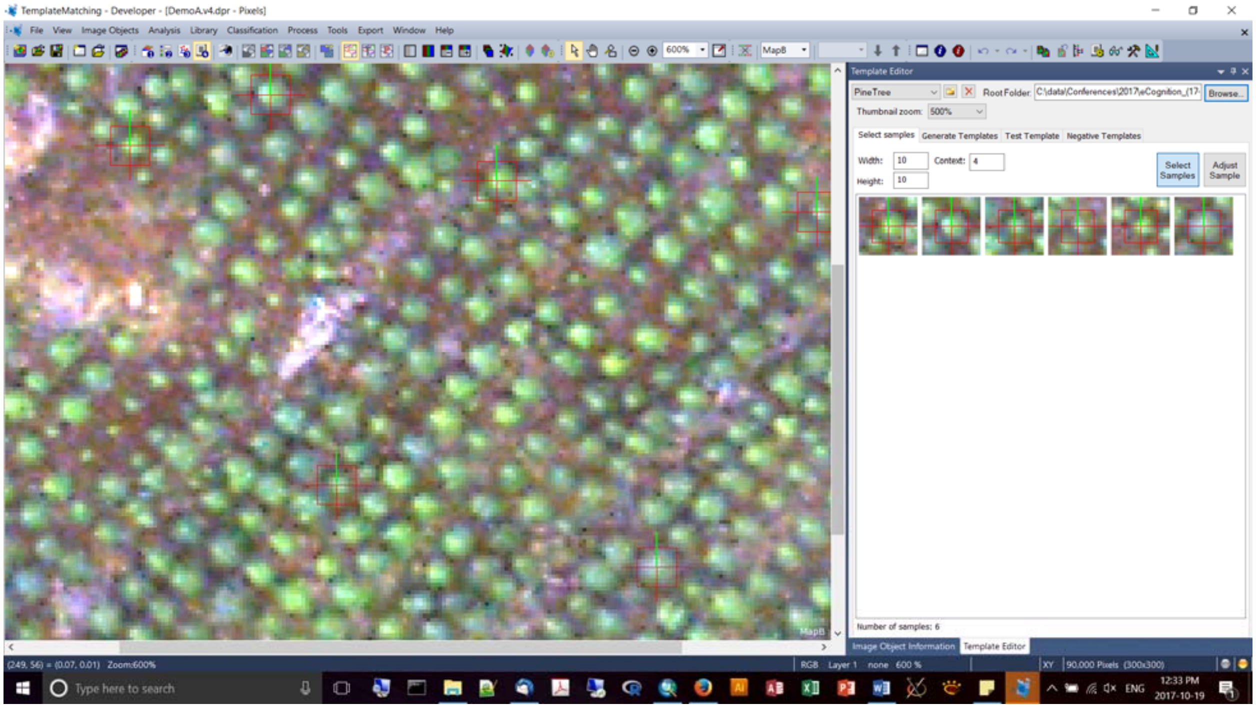

TemplateMatching - Developer - [DemoA.v4.dpr - Pixels]
File View Image Objects Analysis Library Classification Process Tools Export Window Help
300%
main
main
MapB
Workspace
TemplateMatching
Name State Scale Type
DemoA Edited 0.000... Origi...
List View (default) Refresh Filters 1 items
Summary
Scene name: DemoA
State: Edited
Time:
Remarks:
Export Specification
C:\data\Conferences\2017\eCognition_(17-19Oct2017)
main
Select active map.
RGB Layer 1 none 300 %
XY 90,000 Pixels (300x300)
Type here to search
ENG 12:17 PM 2017-10-19

- Right click anywhere on the menu bar and launch “Template Editor”



- Good for when objects are of uniform shape and size.

- Create a name and specify a workspace for your template. It's good to have "template" folders within your project to hold these.
- When you center your cross-hair on an example feature, you can specify how big the box should be, and specify how many pixels the boundary should have ("context")



TemplateMatching - Developer - [DemoA.v4.dpr - Pixels]

Template Editor

PineTree Root Folder: C:\data\Conferences\2017\eCognition_(17-

Thumbnail zoom: 500%

Select samples | Generate Templates | Test Template | Negative Templates

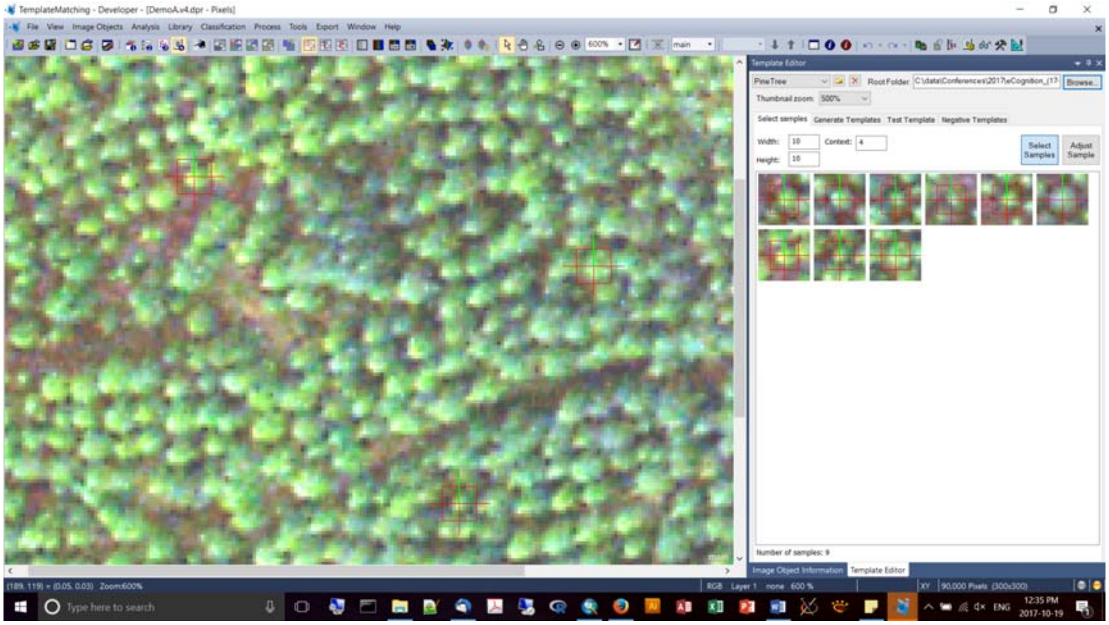
Width: 10 Context: 4

Height: 10

Number of samples: 6

File View Image Objects Analysis Library Classification Process Tools Export Window Help
600%
MapB
Browse...
Select Samples
Adjust Sample
Image Object Information
Template Editor
(249, 56) = (0.07, 0.01) Zoom:600%
RGB Layer 1 none 600 %
XY 90,000 Pixels (300x300)
Type here to search
12:33 PM
2017-10-19

- Now I toggle to the other map; notice how the samples are getting pooled, though they're from different maps:



TemplateMatching - Developer - [DemoA.v4.dpr - Pixels]

Template Editor

PineTree Root Folder: C:\data\Conferences\2017\eCognition_(17-

Thumbnail zoom: 500%

Select samples | Generate Templates | Test Template | Negative Templates

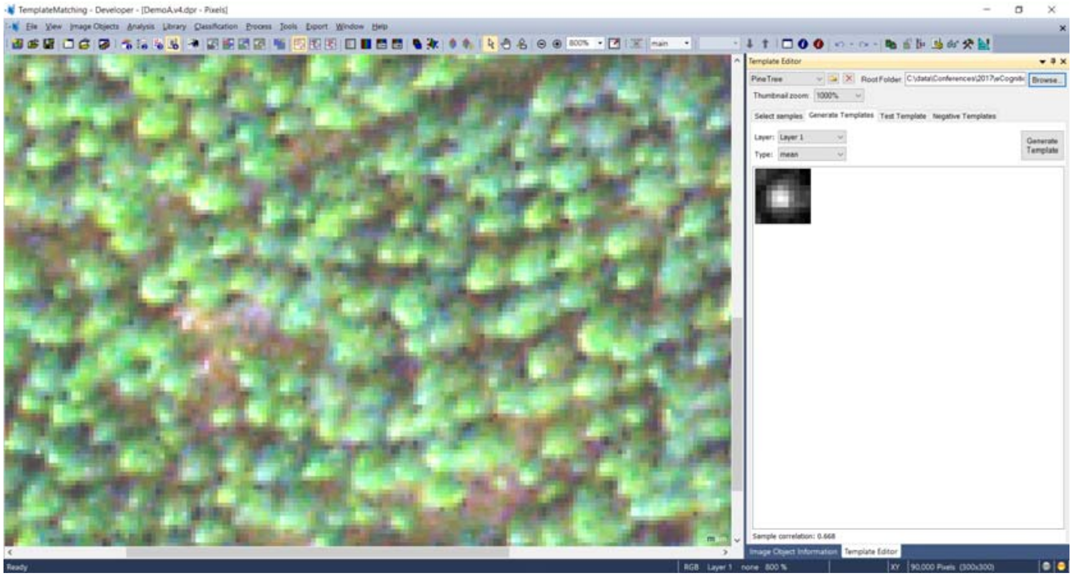
Width: 10 Context: 4

Height: 10

Number of samples: 9

File View Image Objects Analysis Library Classification Process Tools Export Window Help
600%
main
Browse...
Select Samples
Adjust Sample
Image Object Information
Template Editor
(189, 119) = (0.05, 0.03) Zoom:600%
RGB Layer 1 none 600 %
XY 90,000 Pixels (300x300)
Type here to search
12:35 PM
2017-10-19

- You can also select to “Adjust the Sample”, which allows you to modify the bounding box and orientation.
- In the next step, you can generate the template, specifying which band (layer) you want to use. N.B. you can use mean pixel values, with a mask, and grouped to allow you to capture some variation in your sample.
- Once you generate your template, it is a mean composite of all of your samples:



PineTree

Root Folder: C:\data\Conferences\2017\eCognit

Thumbnail zoom: 1000%

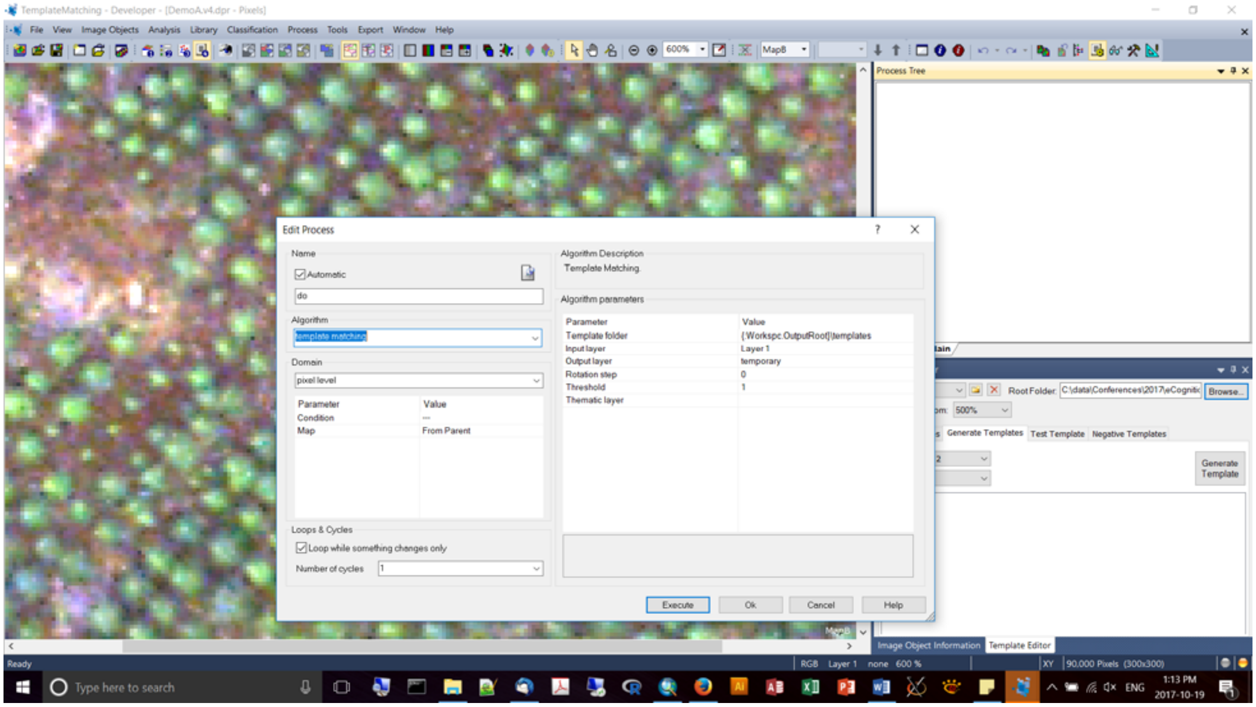
Layer: Layer 1

Type: mean

Sample correlation: 0.668

- Notice the level of correlation statistic.
- Use the layer/template (green band) with the highest correlation between samples.
- You’ll notice that the templates have now been stored as .tiffs in the folder you specified.
- Then move into the tab “Test Template”
- You have other parameters to set: Ground truth tolerance, threshold (use the same value as your correlation statistics), and Rot. Step (if you want to specify increments of rotation for your template matching if your object isn’t round – the smaller the number the longer the processing time).
- If you lower the threshold it’s going to pick up trees that don’t quite match your template; but be careful, you can produce more false positives.
- By drawing a bounding box it will pull objects for you, using your template.
- You can inspect each of the “unclassified” objects and indicate whether it’s “Correct” or “False”. Objects then get moved into either the “Correct” or “False” bins depending, and you will get statistics on the quality of your template.
- Notice the checkbox for “Update Template” – You can also go back to “Generate Templates” and click again on the “Generate template” button to regenerate your template. My correlation went up to 0.877.

- Setting up a template matching process:



Edit Process

Name

☑ Automatic

do

Algorithm: template matching

Domain: pixel level

Parameter	Value
Condition	---
Map	From Parent

Loops & Cycles

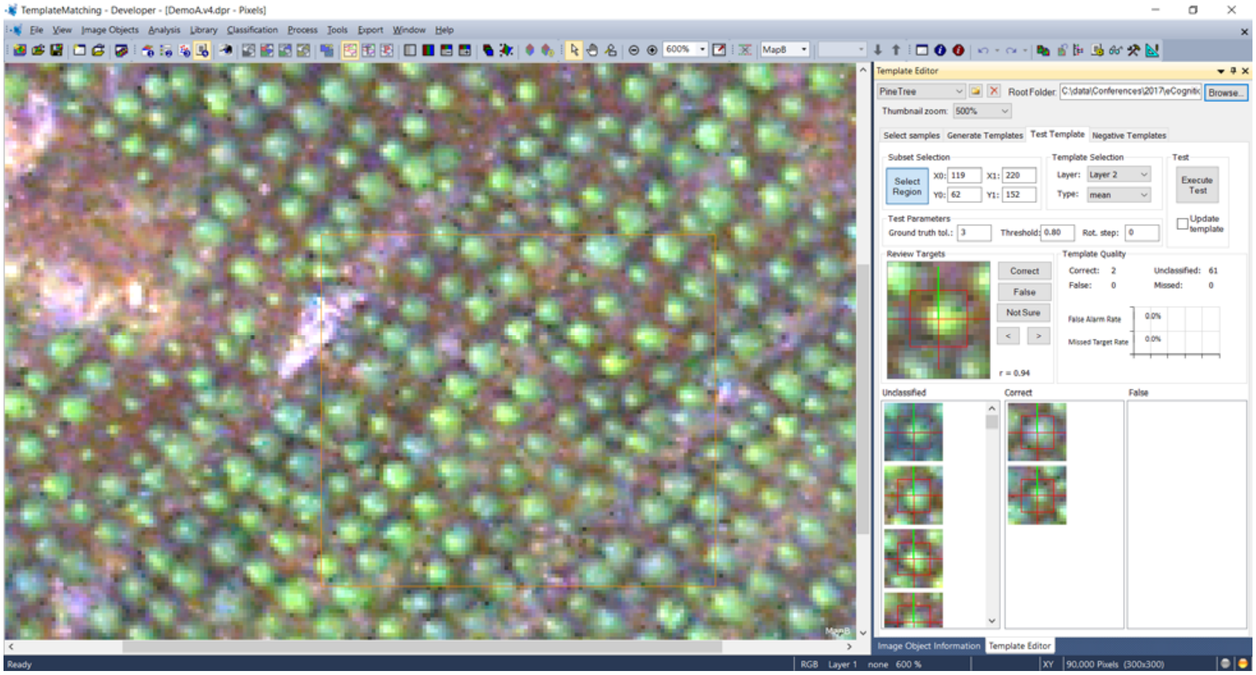
☑ Loop while something changes only

Number of cycles: 1

Algorithm Description: Template Matching.

Algorithm parameters

Parameter	Value
Template folder	{:Workspc.OutputRoot}\templates
Input layer	Layer 1
Output layer	temporary
Rotation step	0
Threshold	1
Thematic layer	



Template Editor

PineTree — Root Folder: C:\data\Conferences\2017\eCognitic

Thumbnail zoom: 500%

Test Template

Subset Selection: X0: 119 X1: 220 Y0: 62 Y1: 152

Template Selection: Layer: Layer 2 Type: mean

Test Parameters: Ground truth tol.: 3 Threshold: 0.80 Rot. step: 0

□ Update template

Template Quality: Correct: 2 Unclassified: 61 False: 0 Missed: 0

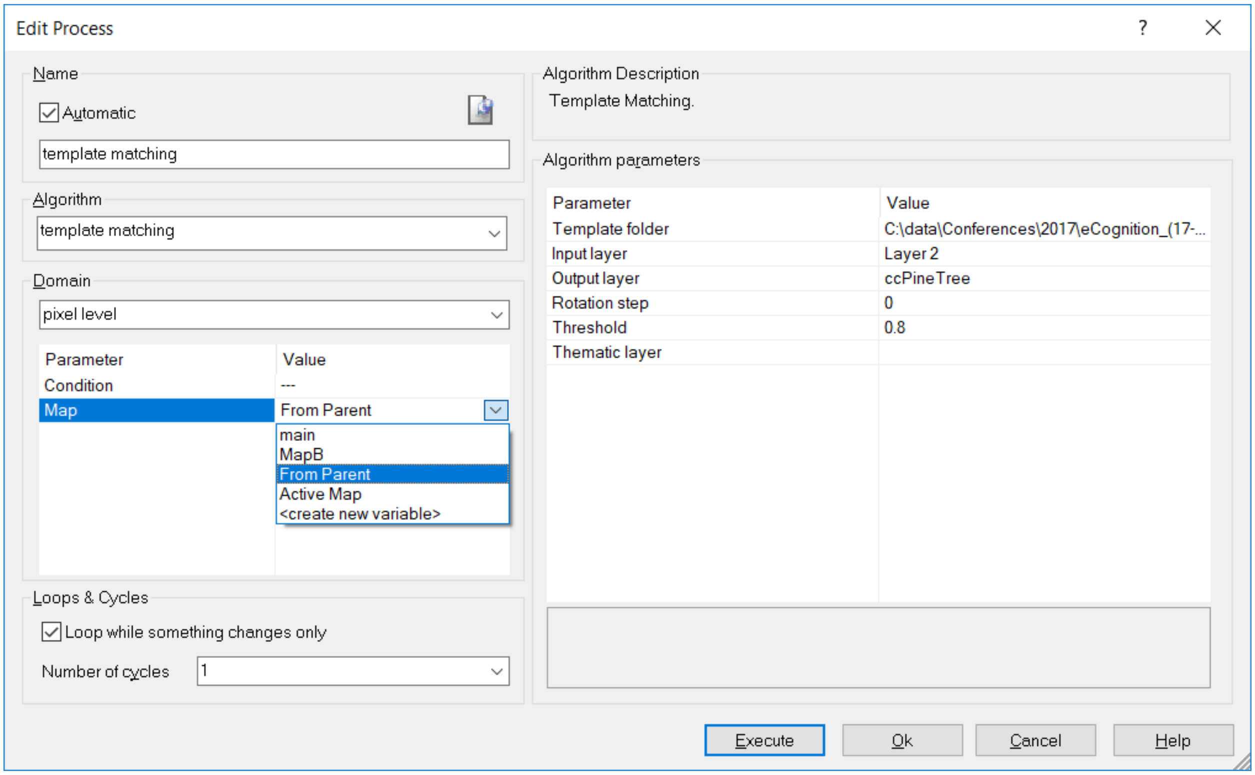
False Alarm Rate 0.0%

Missed Target Rate 0.0%

r = 0.94

- The following illustrates how you need to set the template folder to the folder you created during your template matching procedure (in our case, the folder with mean layer 2).
- Input layer is also set to “Layer 2” (i.e., band 2).

- Note which map layer do you want to apply to?



Edit Process

Name

☑ Automatic

template matching

Algorithm

template matching

Domain

pixel level

Parameter	Value
Condition	---
Map	From Parent

Loops & Cycles

☑ Loop while something changes only

Number of cycles 1

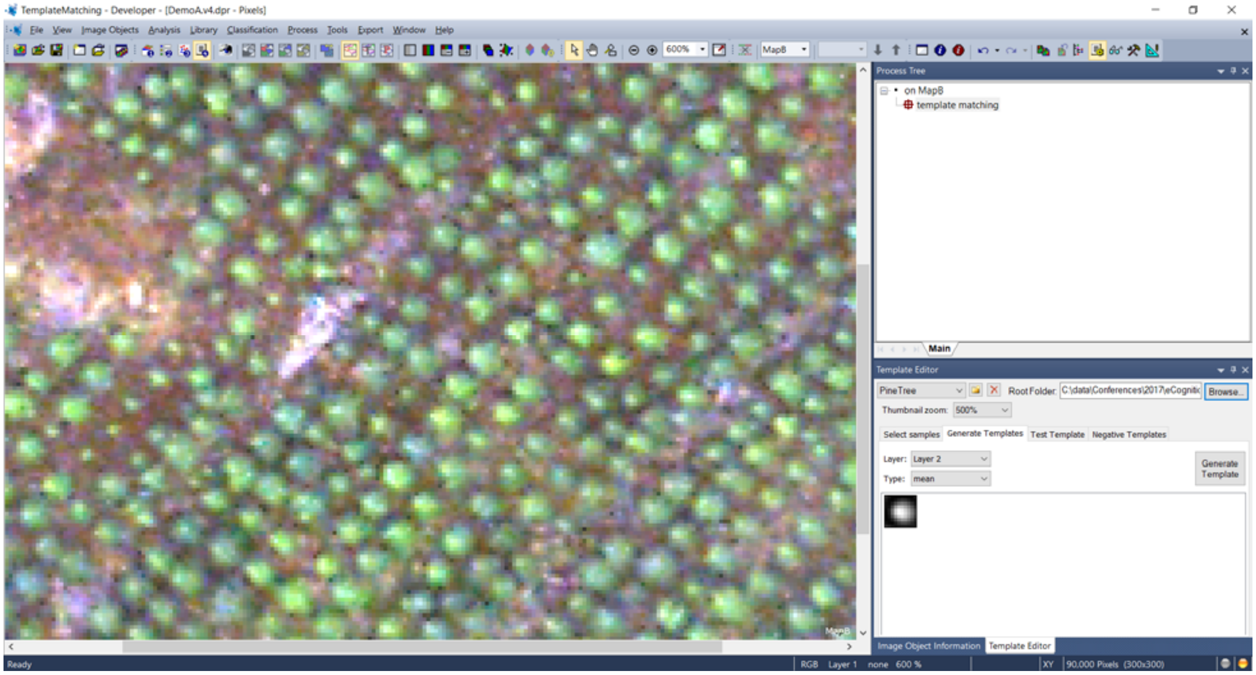
Algorithm Description

Template Matching.

Algorithm parameters

Parameter	Value
Template folder	C:\data\Conferences\2017\eCognition_(17-...
Input layer	Layer 2
Output layer	ccPineTree
Rotation step	0
Threshold	0.8
Thematic layer	

- The following illustrates how you can embed the map layer you are processing – add a new process, and choose the map to be “MapB” (or whatever). Now when you embed a child process, you can specify “from parent” in the



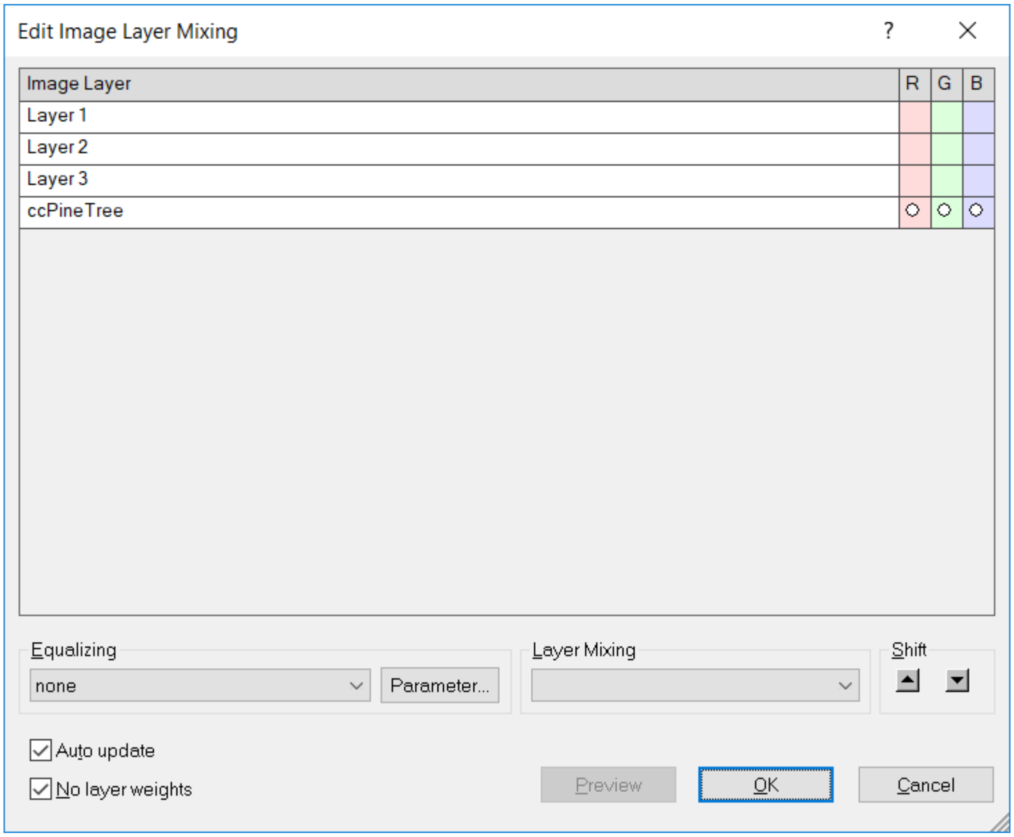
PineTree Root Folder: C:\data\Conferences\2017\eCognitic

Thumbnail zoom: 500%

Layer: Layer 2

Type: mean

- To display the features, choose to Edit Image Laery Mixing



Edit Image Layer Mixing

Image Layer	R	G	B
Layer 1			
Layer 2			
Layer 3			
ccPineTree	○	○	○

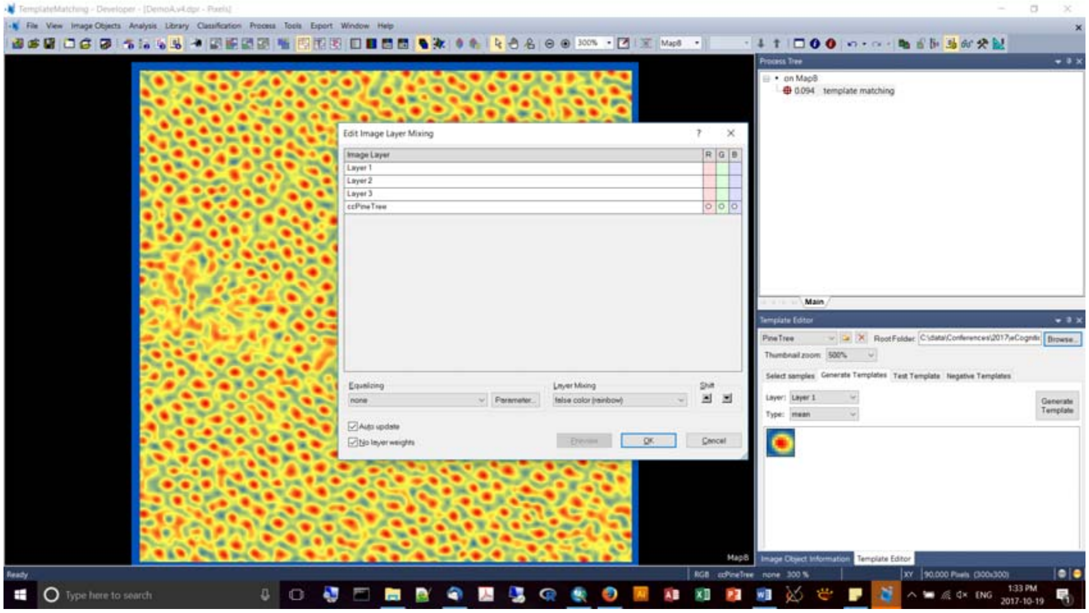
Equalizing: none

Layer Mixing:

☑ Auto update

☑ No layer weights

- Another way to display (false colour rainbow):



TemplateMatching - Developer - [DemoA.v4.dpr - Pixels]
File View Image Objects Analysis Library Classification Process Tools Export Window Help
300%
Map8
Process Tree

on Map8
0.094 template matching

Edit Image Layer Mixing

Image Layer	R	G	B
Layer 1			
Layer 2			
Layer 3			
ccPineTree	○	○	○

Equalizing: none

Layer Mixing: false color (rainbow)

☑ Auto update

☑ No layer weights

PineTree Root Folder: C:\data\Conferences\2017\eCognit

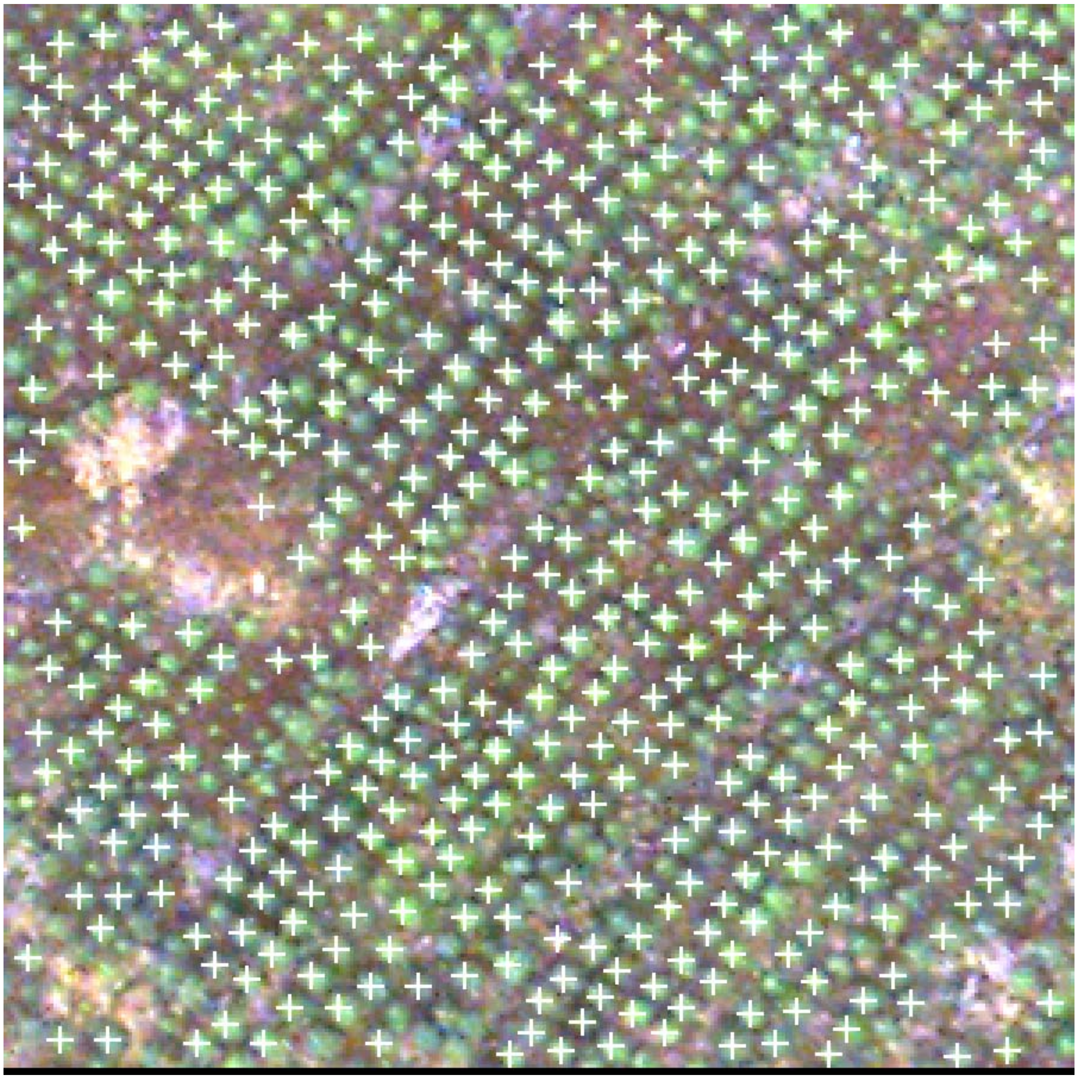
Thumbnail zoom: 500%

Layer: Layer 1

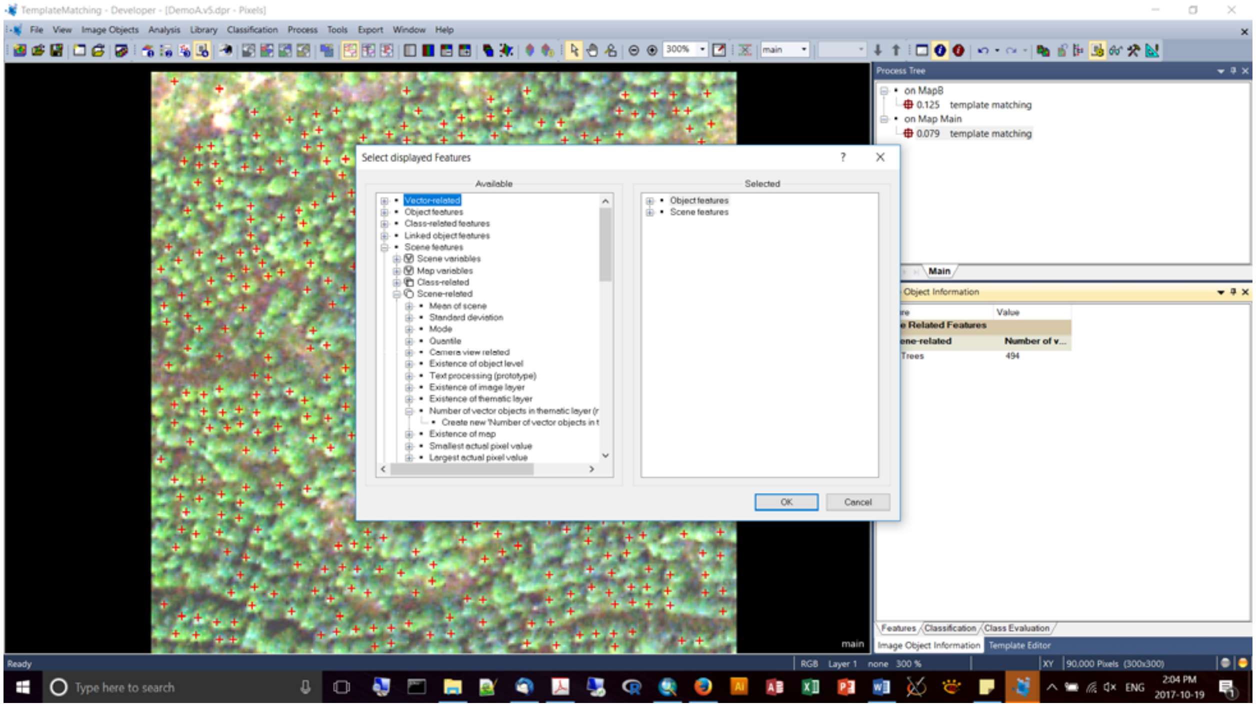
Type: mean

RGB ccPineTree none 300 % XY 90,000 Pixels (300x300)

- You can specify a Thematic Layer to produce vector points for output
- You can add the points as a vector layer for inspection:

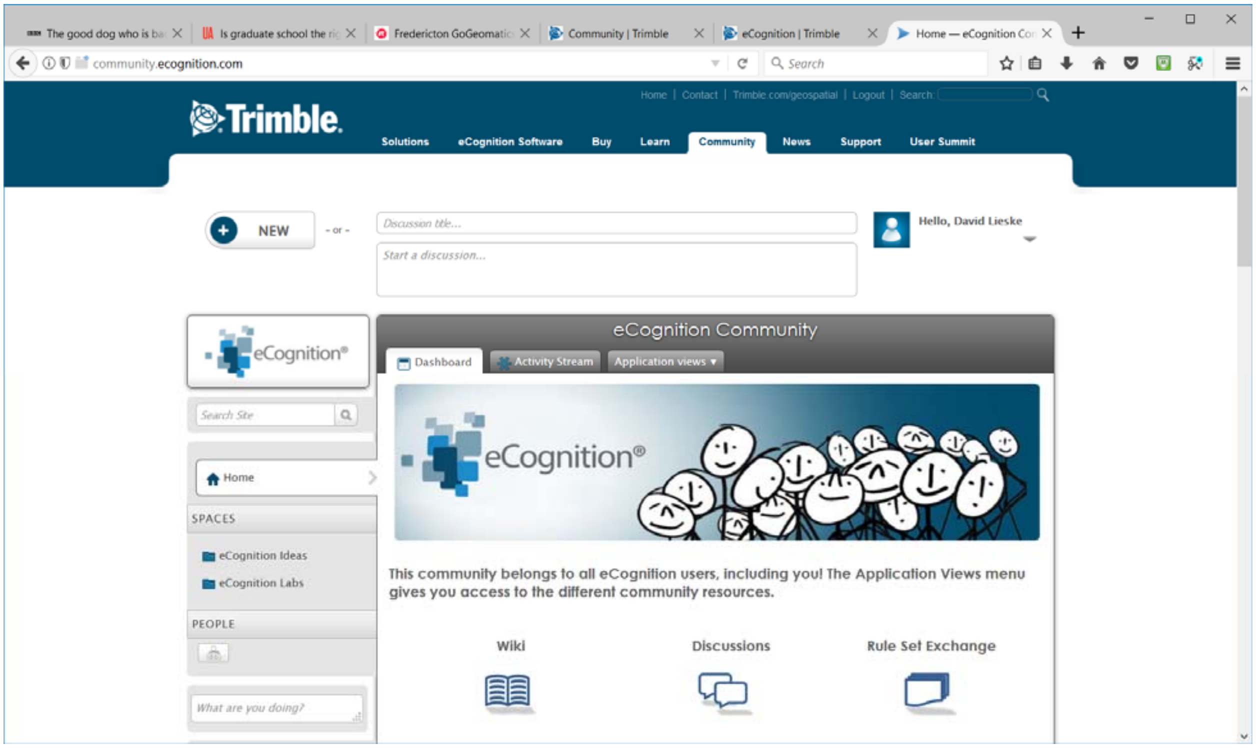


- How do we count the number of trees?
- You can choose the image object information button (the blue “i”) and add the scene level summary “Number of vector objects in thematic layer” to produce a total count of vector objects in the display.
- In this case, there are 494 trees.



- You can check out more examples on the eCognition Community link:

  http://www.ecognition.com/community



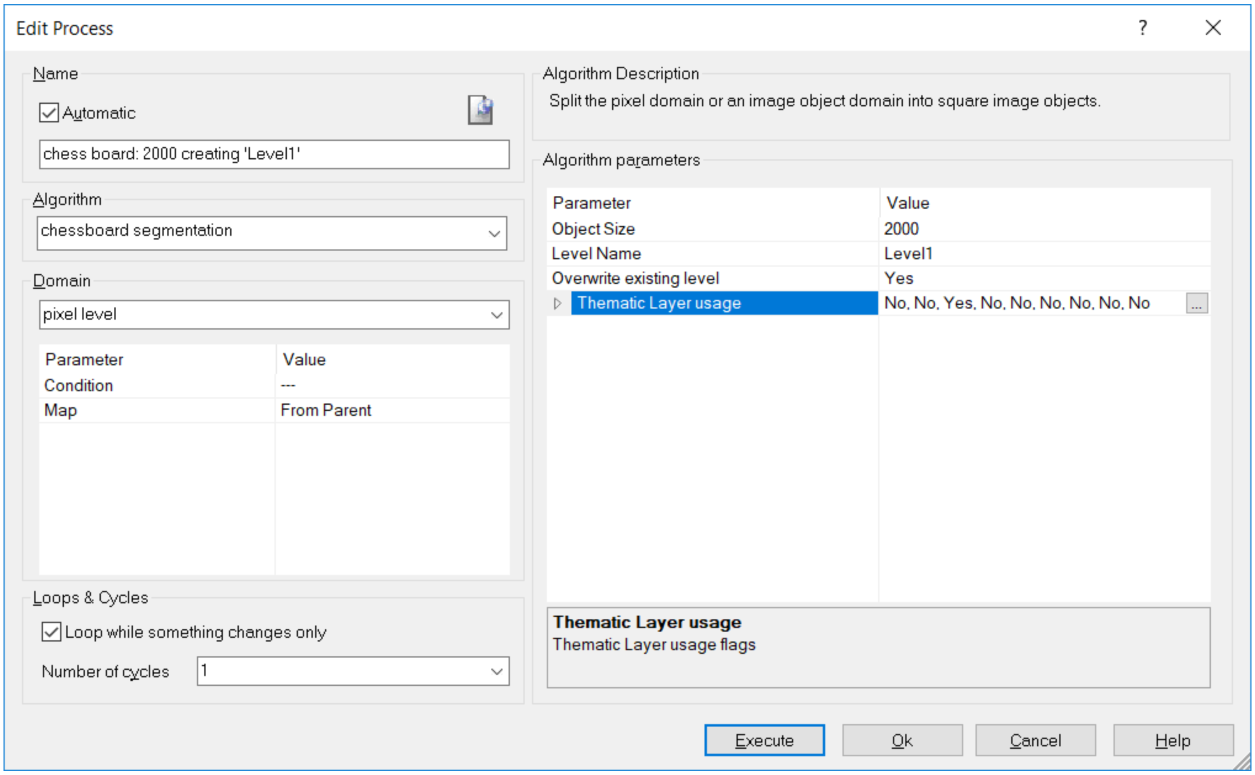
This community belongs to all eCognition users, including you! The Application Views menu gives you access to the different community resources.

POINTS CLOUDS IN ECOGNITION

- When you bring point clouds into eCog it produces rasters from it based on intensity.
- Ex., extracting a DSM layer from a Lidar point cloud.
- Algorithim: rasterize point cloud.
- Resolving low data sections via gap filling: (1) use a segmentation process to identify the unclassified pixles, (2) apply a “Median filter” to apply the neighbourhood values to your unclassified pixels.
- Keith: when we release eCog 9.3 there will be tutorials on point cloud classification, and point cloud change detection tutorials.
- Any plan to apply templates to point clouds?

Exercise #4 – Urban Building Footprint Change, with thematic-layer based segmentation and classification, with segmentation and classification, doing vector-based object refinement, and finally, GIS-ready export

- For example: cadastral data is generally not 100% up to date. Freshly collected imagery will have buildings not “known” by the cadastre.
- We’re work with 4-band VHR data, Lidar based DSM, DTM, and a building footprint (.shp)
- For this exercise you’ll load in a pre-existing rule set (look for a *.dcp).
- Goal is to find new buildings. What will help us out? Elevation, NDVI, height (ndsm), vector layer will help us to determine what’s “new”. The following is the procedure for creating a chessboard focusing on the vector data:



☑ Automatic

chess board: 2000 creating 'Level1'

Algorithm: chessboard segmentation

Domain: pixel level

Parameter	Value
Condition	---
Map	From Parent

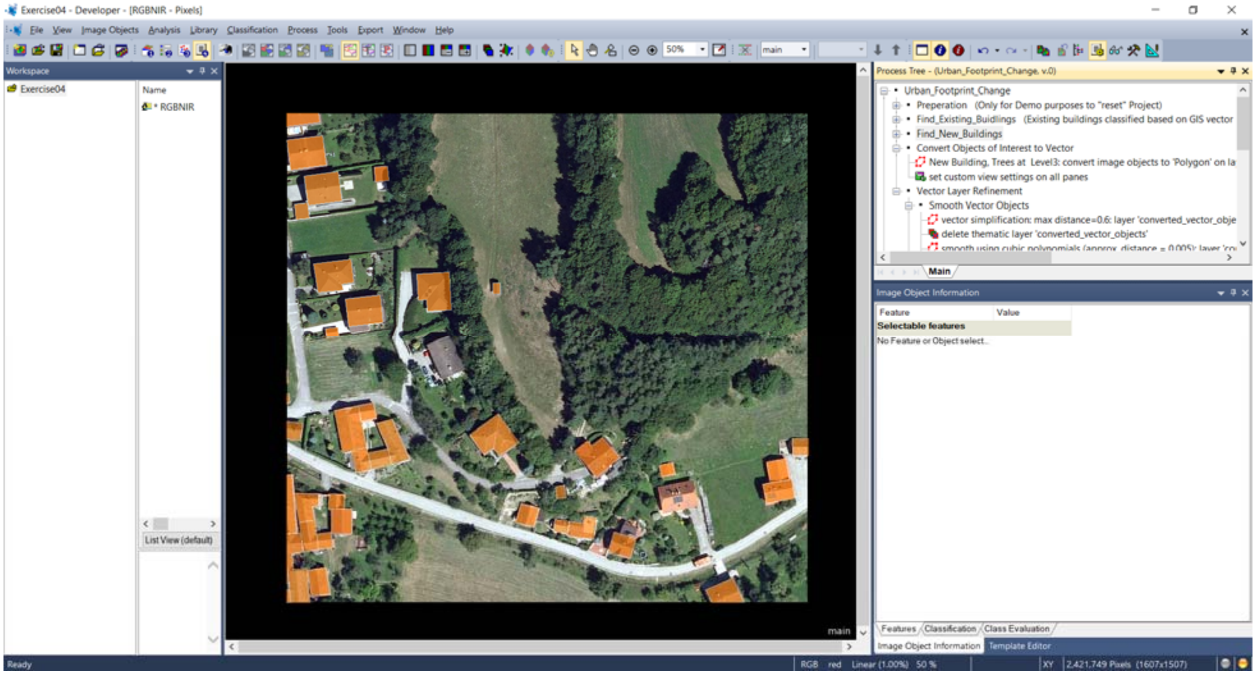
☑ Loop while something changes only

Number of cycles: 1

Algorithm Description: Split the pixel domain or an image object domain into square image objects.

Parameter	Value
Object Size	2000
Level Name	Level1
Overwrite existing level	Yes
Thematic Layer usage	No, No, Yes, No, No, No, No, No, No

Thematic Layer usage
Thematic Layer usage flags

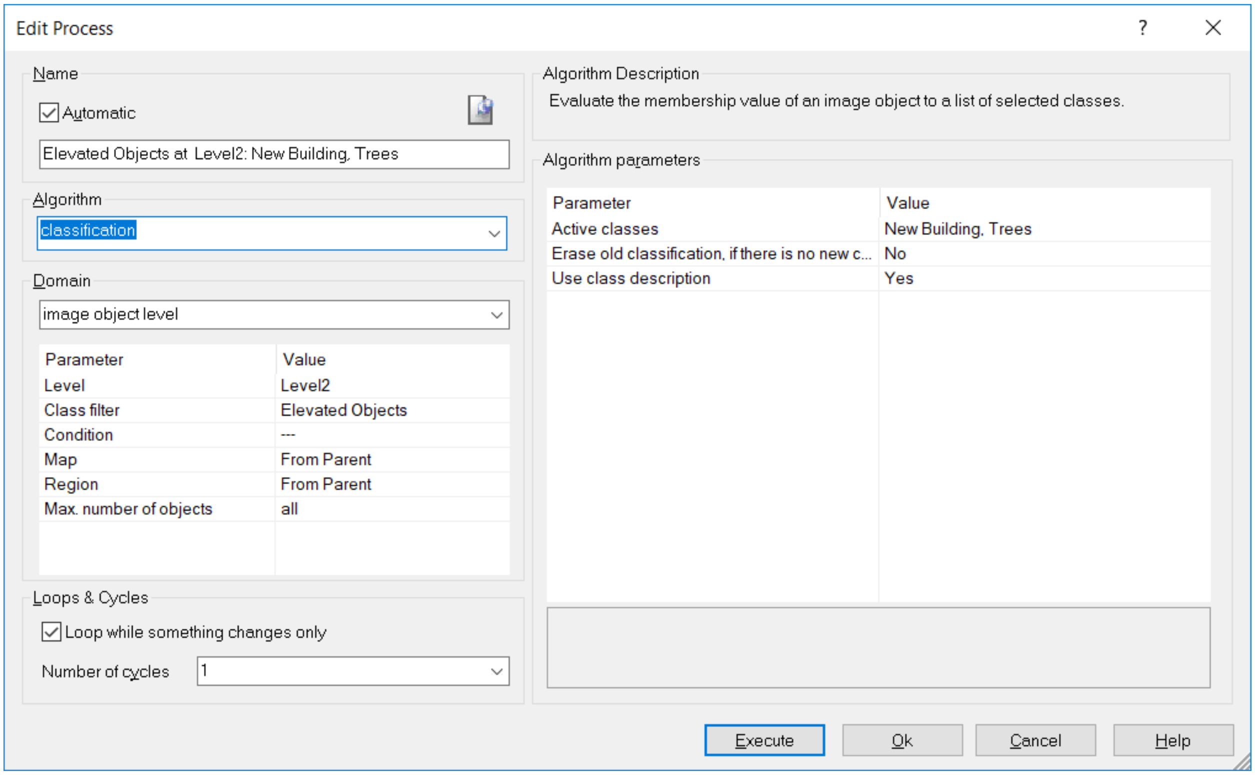


- Urban_Footprint_Change
  - Preperation (Only for Demo purposes to "reset" Project)
  - Find_Existing_Buidlings (Existing buildings classified based on GIS vector
  - Find_New_Buildings
  - Convert Objects of Interest to Vector
    - New Building, Trees at Level3: convert image objects to 'Polygon' on la
    - set custom view settings on all panes
  - Vector Layer Refinement
    - Smooth Vector Objects
      - vector simplification: max distance=0.6: layer 'converted_vector_obje
      - delete thematic layer 'converted_vector_objects'
      - smooth using cubic polynomials (approx. distance = 0.005): layer 'co

Feature	Value
Selectable features	
No Feature or Object select...	

- “Assign Class” vs. “**Classification**” = Assign Class allows for a flexible class definition schema.
- But they are restricted to Boolean operators.

- “Classification” has more class description tools. Boolean and membership functions can be used.
- Class definition is in the class itself.



Name

☑ Automatic

Elevated Objects at Level2: New Building, Trees

Algorithm

classification

Domain

image object level

Parameter	Value
Level	Level2
Class filter	Elevated Objects
Condition	---
Map	From Parent
Region	From Parent
Max. number of objects	all

Loops & Cycles

☑ Loop while something changes only

Number of cycles 1

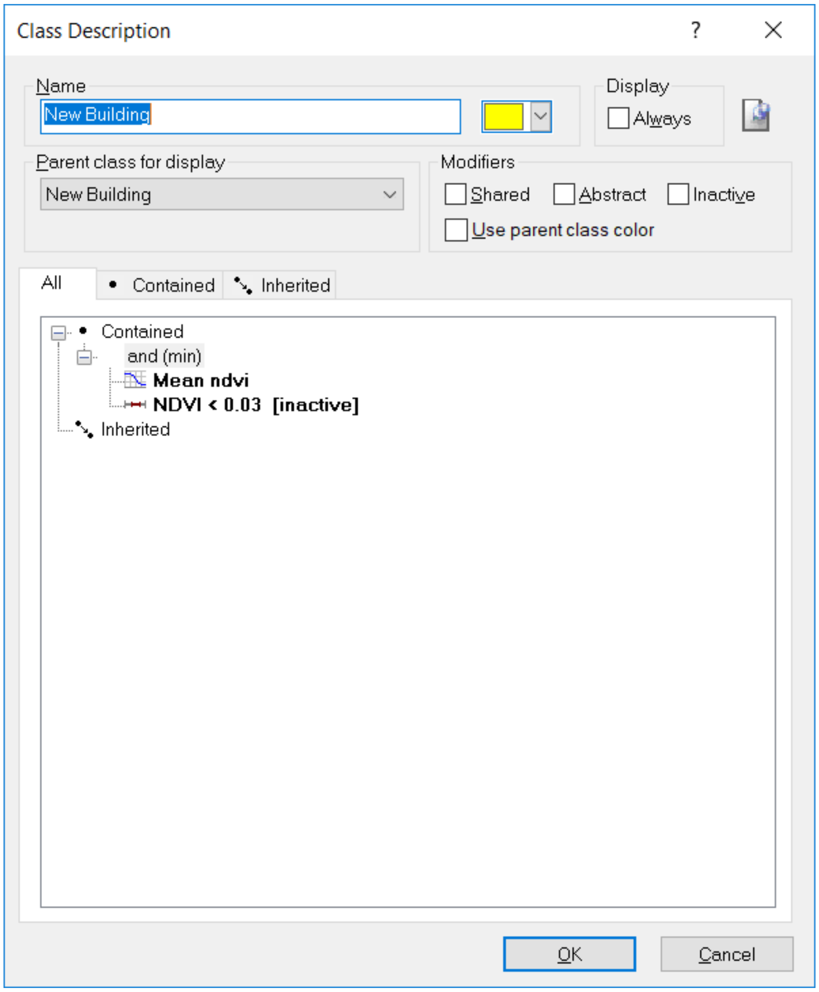
Algorithm Description

Evaluate the membership value of an image object to a list of selected classes.

Algorithm parameters

Parameter	Value
Active classes	New Building, Trees
Erase old classification, if there is no new c...	No
Use class description	Yes

- Look at the class in the class hierarchy – there are rules defining membership, which the Classification algorithm will access. The class defifintion is fixed. You can also flexibly define how the class membership is assigned.



Class Description

Name: New Building

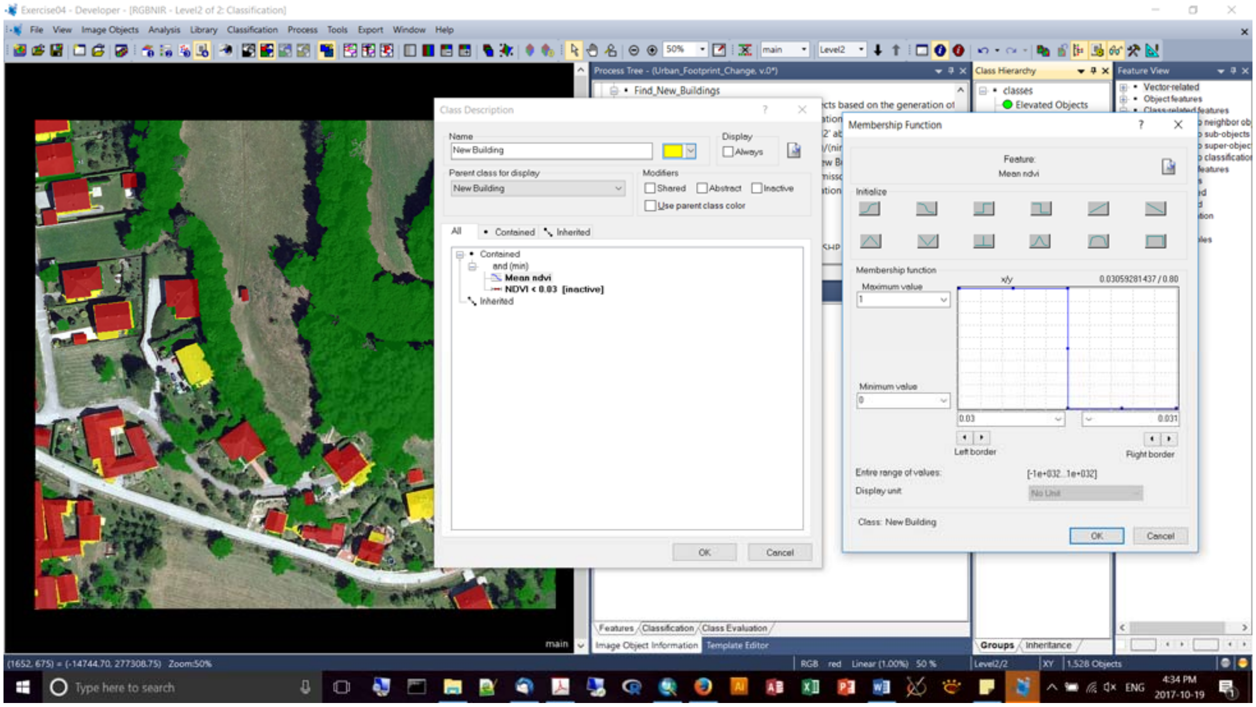
Display: □ Always

Parent class for display: New Building

Modifiers: □ Shared □ Abstract □ Inactive □ Use parent class color

All | • Contained | Inherited

```
⊟ • Contained
    ⊟ and (min)
        Mean ndvi
        NDVI < 0.03  [inactive]
  Inherited
```



Exercise04 - Developer - [RGBNIR - Level2 of 2: Classification]

Process Tree - (Urban_Footprint_Change, v.0*)
- Find_New_Buildings

Class Hierarchy
- classes
  - Elevated Objects

Class Description

Name: New Building

Parent class for display: New Building

Display: □ Always

Modifiers: □ Shared □ Abstract □ Inactive □ Use parent class color

- Contained
  - and (min)
    - **Mean ndvi**
    - **NDVI < 0.03 [inactive]**
- Inherited

Membership Function

Feature: Mean ndvi

Maximum value: 1

Minimum value: 0

x/y 0.03059281437 / 0.80

Left border: 0.03

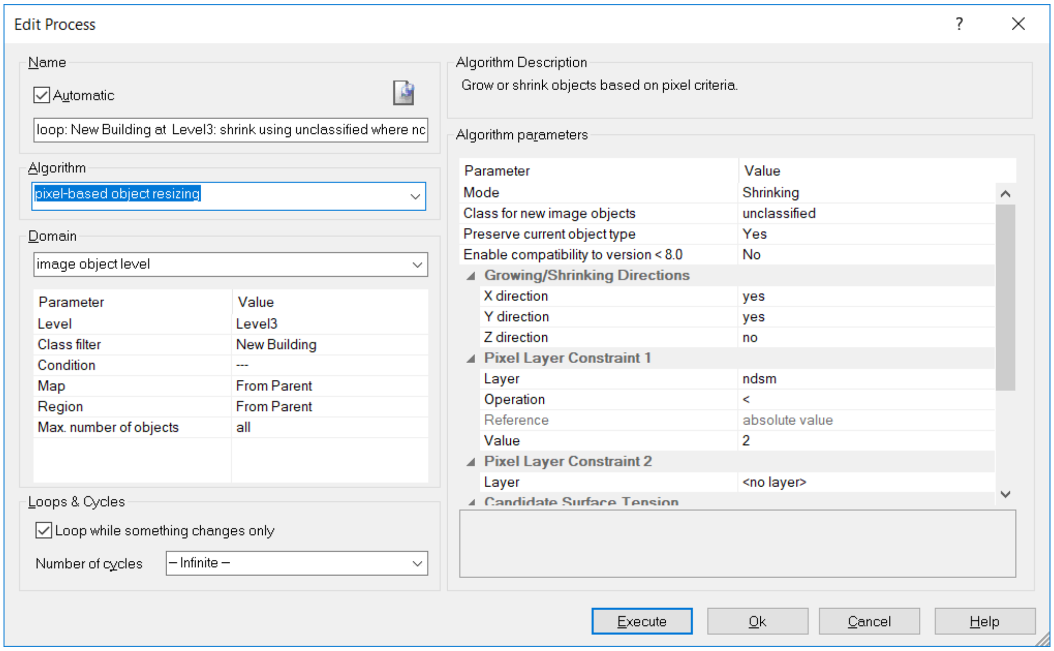
Right border: 0.031

Entire range of values: [-1e+032..1e+032]

Display unit: No Unit

Class: New Building

- We are going to use a new algorithm: **pixel-based object resizing**
- We're using mode "Shrinking". This is because there are some erroneous "threads" streaming off these buildings.
- When we remove pixels, they get assigned to the "unclassified" class.
- Loops and cycles are also important for the execution of this algorithm.



Edit Process

Name

☑ Automatic

loop: New Building at Level3: shrink using unclassified where nc

Algorithm

pixel-based object resizing

Domain

image object level

Parameter	Value
Level	Level3
Class filter	New Building
Condition	---
Map	From Parent
Region	From Parent
Max. number of objects	all

Loops & Cycles

☑ Loop while something changes only

Number of cycles: – Infinite –

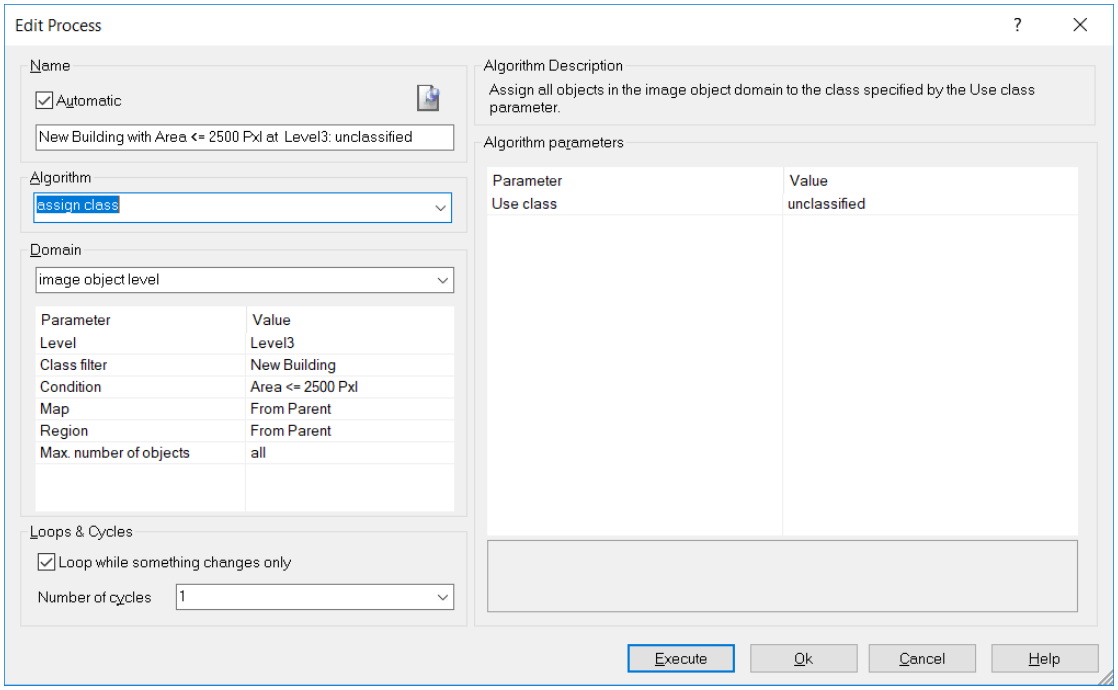
Algorithm Description

Grow or shrink objects based on pixel criteria.

Algorithm parameters

Parameter	Value
Mode	Shrinking
Class for new image objects	unclassified
Preserve current object type	Yes
Enable compatibility to version < 8.0	No
Growing/Shrinking Directions	
X direction	yes
Y direction	yes
Z direction	no
Pixel Layer Constraint 1	
Layer	ndsm
Operation	<
Reference	absolute value
Value	2
Pixel Layer Constraint 2	
Layer	<no layer>
Candidate Surface Tension	

- Keith then went and threw away the small buildings (assigning them back into the "unclassified" category):



☑ Automatic

New Building with Area <= 2500 Pxl at Level3: unclassified

Algorithm: assign class

Domain: image object level

Parameter	Value
Level	Level3
Class filter	New Building
Condition	Area <= 2500 Pxl
Map	From Parent
Region	From Parent
Max. number of objects	all

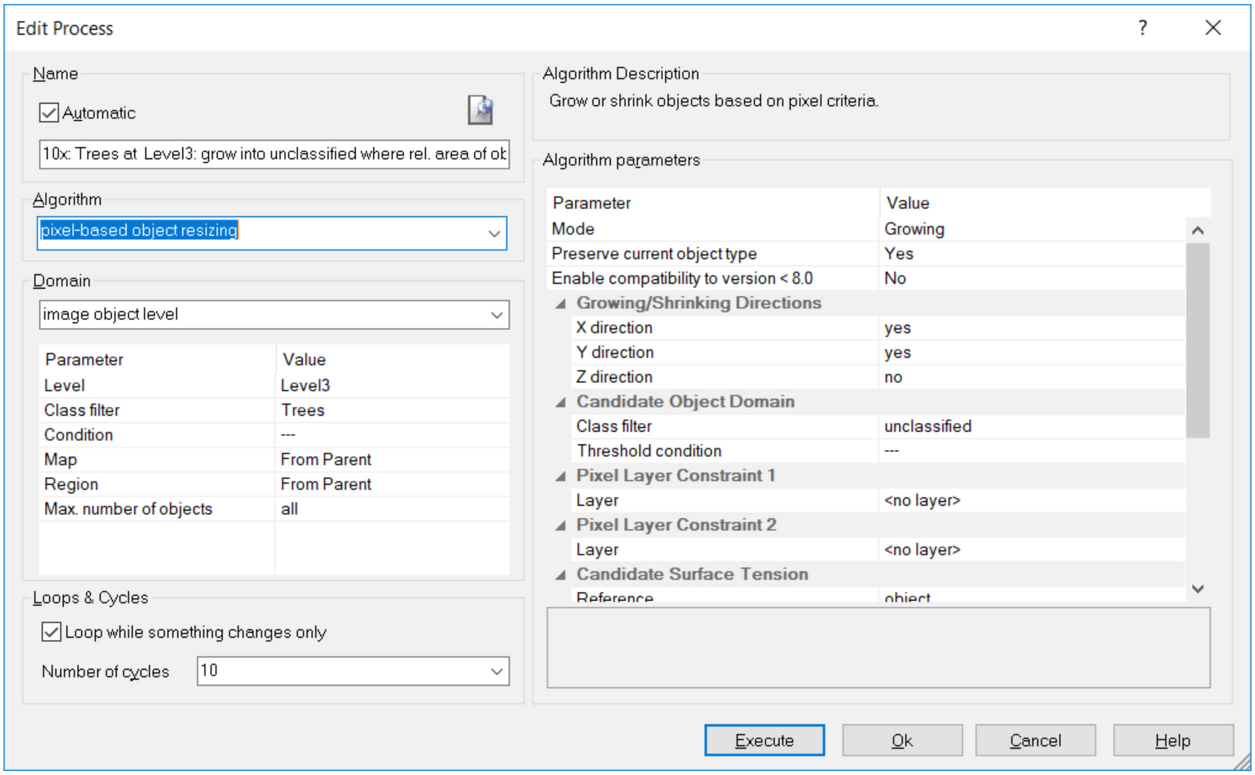
Algorithm Description: Assign all objects in the image object domain to the class specified by the Use class parameter.

Parameter	Value
Use class	unclassified

☑ Loop while something changes only

Number of cycles: 1

- Keith also applies a grow with a reference made to the surface tension:



Edit Process

Name

☑ Automatic

10x: Trees at Level3: grow into unclassified where rel. area of ob

Algorithm

pixel-based object resizing

Domain

image object level

Parameter	Value
Level	Level3
Class filter	Trees
Condition	---
Map	From Parent
Region	From Parent
Max. number of objects	all

Loops & Cycles

☑ Loop while something changes only

Number of cycles: 10

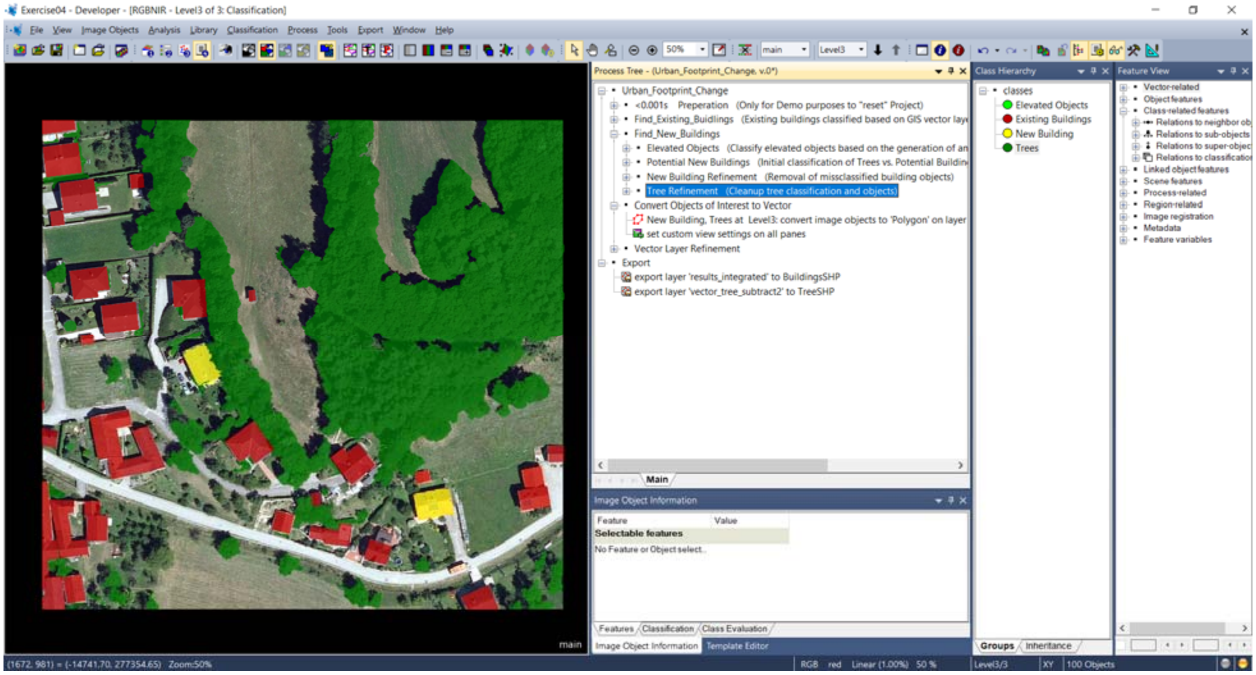
Algorithm Description

Grow or shrink objects based on pixel criteria.

Algorithm parameters

Parameter	Value
Mode	Growing
Preserve current object type	Yes
Enable compatibility to version < 8.0	No
Growing/Shrinking Directions	
X direction	yes
Y direction	yes
Z direction	no
Candidate Object Domain	
Class filter	unclassified
Threshold condition	---
Pixel Layer Constraint 1	
Layer	<no layer>
Pixel Layer Constraint 2	
Layer	<no layer>
Candidate Surface Tension	
Reference	object

- We can convert our objects to vector objects, visible as having a “vectory” icon in the tree. (algorithm = “convert image objects to vector objects”)



Exercise04 - Developer - [RGBNIR - Level3 of 3: Classification]

Process Tree - (Urban_Footprint_Change, v.0*)

```
⊟ • Urban_Footprint_Change
  ⊞ • <0.001s   Preperation   (Only for Demo purposes to "reset" Project)
  ⊞ • Find_Existing_Buidlings   (Existing buildings classified based on GIS vector laye
  ⊟ • Find_New_Buildings
    ⊞ • Elevated Objects   (Classify elevated objects based on the generation of an
    ⊞ • Potential New Buildings   (Initial classification of Trees vs. Potential Buildin
    ⊞ • New Building Refinement   (Removal of missclassified building objects)
    ⊞ • Tree Refinement   (Cleanup tree classification and objects)
  ⊟ • Convert Objects of Interest to Vector
      New Building, Trees at  Level3: convert image objects to 'Polygon' on layer
      set custom view settings on all panes
  ⊞ • Vector Layer Refinement
⊟ • Export
    export layer 'results_integrated' to BuildingsSHP
    export layer 'vector_tree_subtract2' to TreeSHP
```

Class Hierarchy

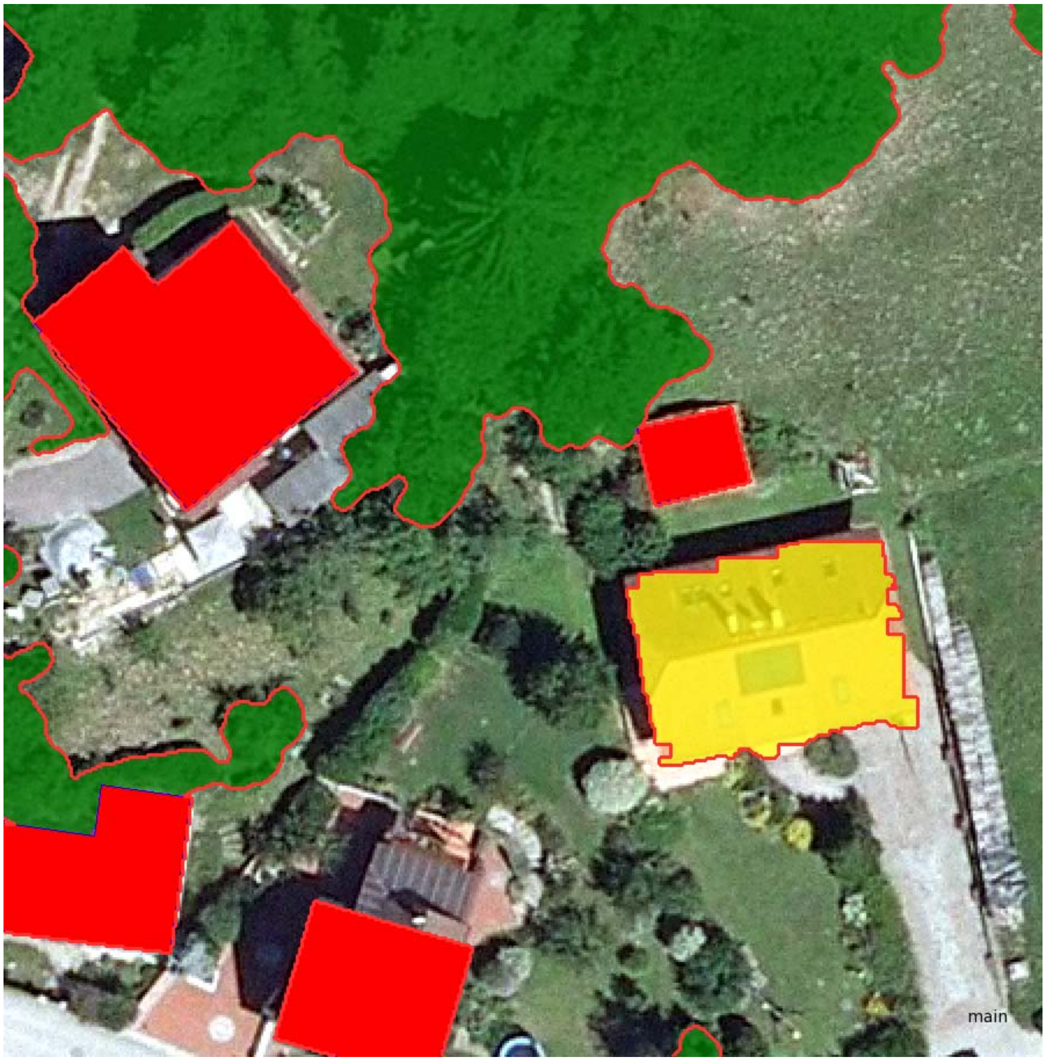
- classes
  - 🟢 Elevated Objects
  - 🔴 Existing Buildings
  - 🟡 New Building
  - 🟢 Trees

Feature View

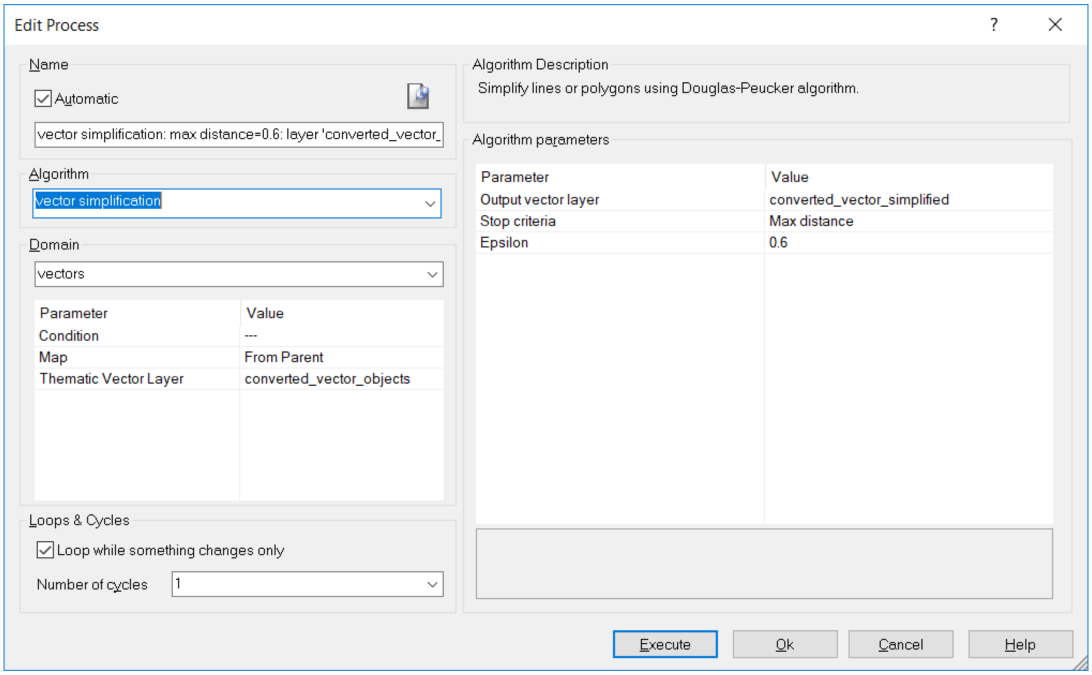
- Vector-related
- Object features
- Class-related features
  - Relations to neighbor ob
  - Relations to sub-objects
  - Relations to super-objec
  - Relations to classificatio
- Linked object features
- Scene features
- Process-related
- Region-related
- Image registration
- Metadata
- Feature variables

Image Object Information

Feature	Value
Selectable features	
No Feature or Object select..	



- Vector simplification:



Name

☑ Automatic

vector simplification: max distance=0.6: layer 'converted_vector_

Algorithm

vector simplification

Domain

vectors

Parameter	Value
Condition	---
Map	From Parent
Thematic Vector Layer	converted_vector_objects

Loops & Cycles

☑ Loop while something changes only

Number of cycles 1

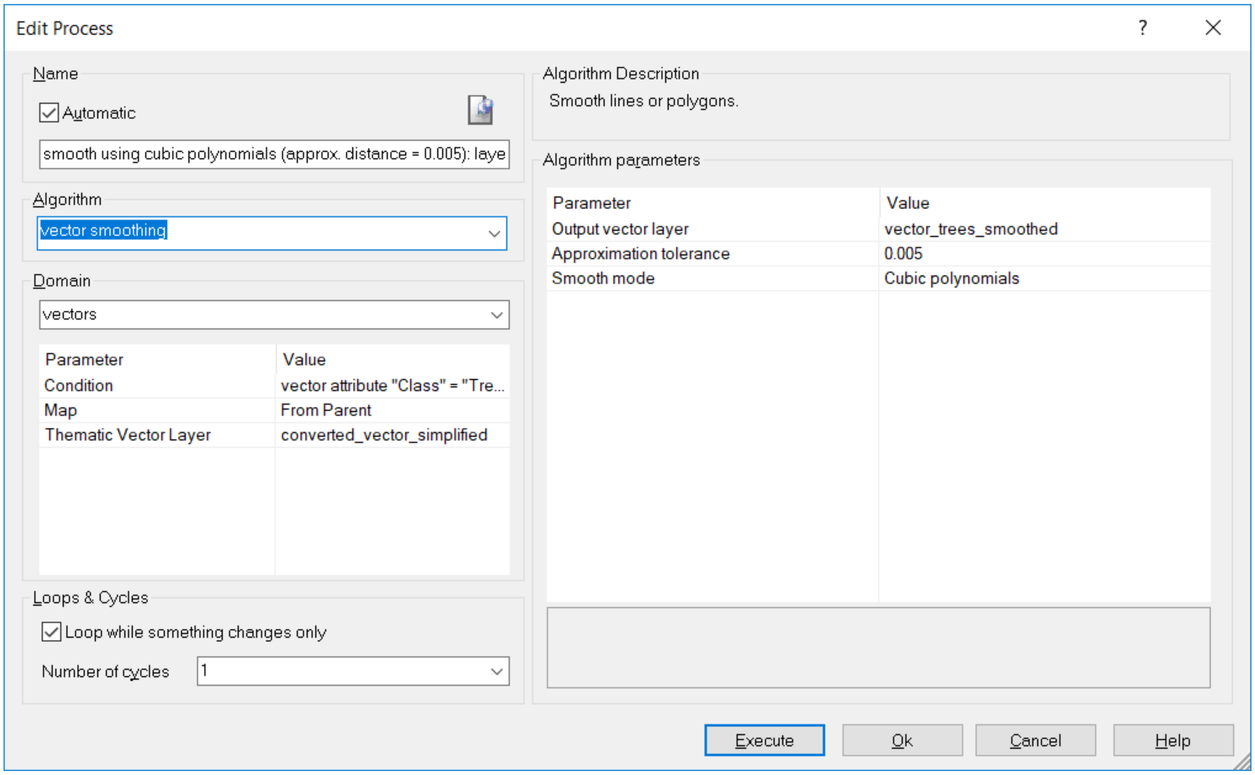
Algorithm Description

Simplify lines or polygons using Douglas-Peucker algorithm.

Algorithm parameters

Parameter	Value
Output vector layer	converted_vector_simplified
Stop criteria	Max distance
Epsilon	0.6

- Vector smoothing:



Name

☑ Automatic

smooth using cubic polynomials (approx. distance = 0.005): laye

Algorithm

vector smoothing

Domain

vectors

Parameter	Value
Condition	vector attribute "Class" = "Tre...
Map	From Parent
Thematic Vector Layer	converted_vector_simplified

Loops & Cycles

☑ Loop while something changes only

Number of cycles 1

Algorithm Description

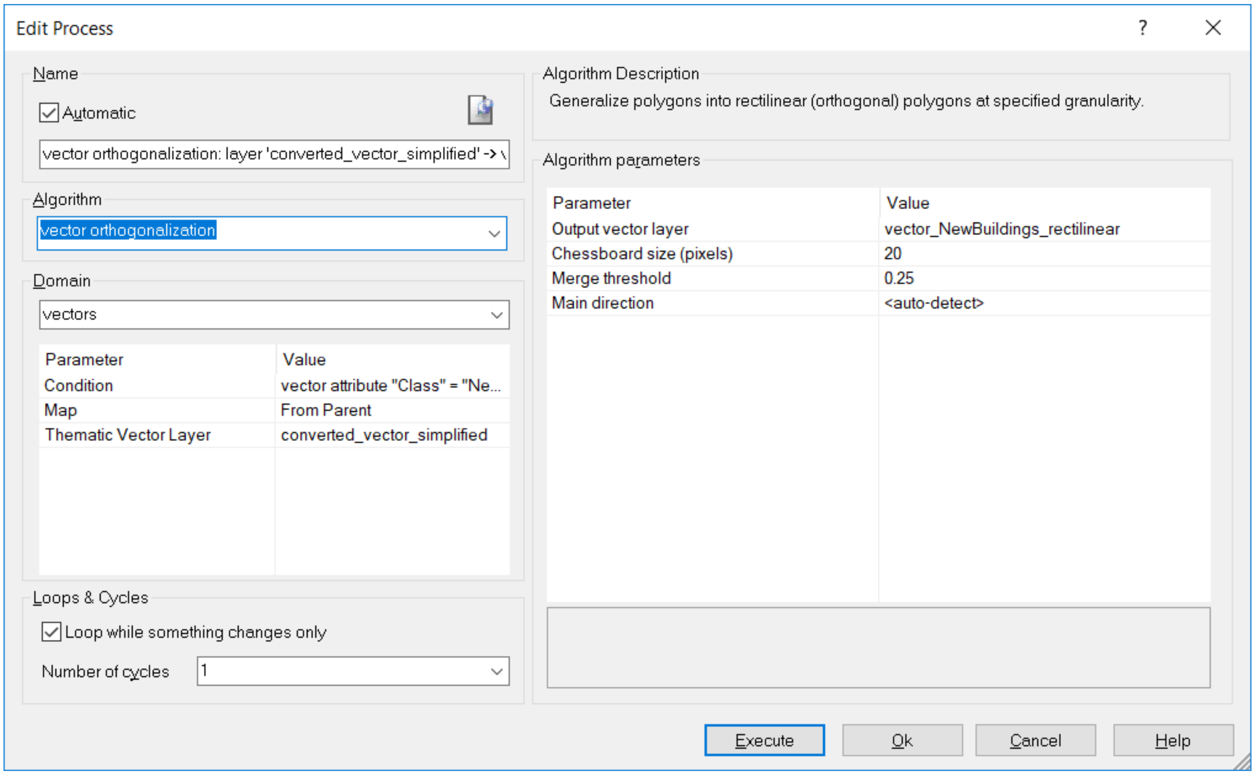
Smooth lines or polygons.

Algorithm parameters

Parameter	Value
Output vector layer	vector_trees_smoothed
Approximation tolerance	0.005
Smooth mode	Cubic polynomials

- Vector orthogonalization = “squaring”

- It’s going to take each building object and put a bounding box around it.
- Within the bounding box we apply a chessboard segmentation of some value, e.g., 0.20.
- The merge threshold determines whether the perimeter will be extended out to the boundary box perimeter.



Name

☑ Automatic

vector orthogonalization: layer 'converted_vector_simplified' -> \

Algorithm

vector orthogonalization

Domain

vectors

Parameter	Value
Condition	vector attribute "Class" = "Ne...
Map	From Parent
Thematic Vector Layer	converted_vector_simplified

Loops & Cycles

☑ Loop while something changes only

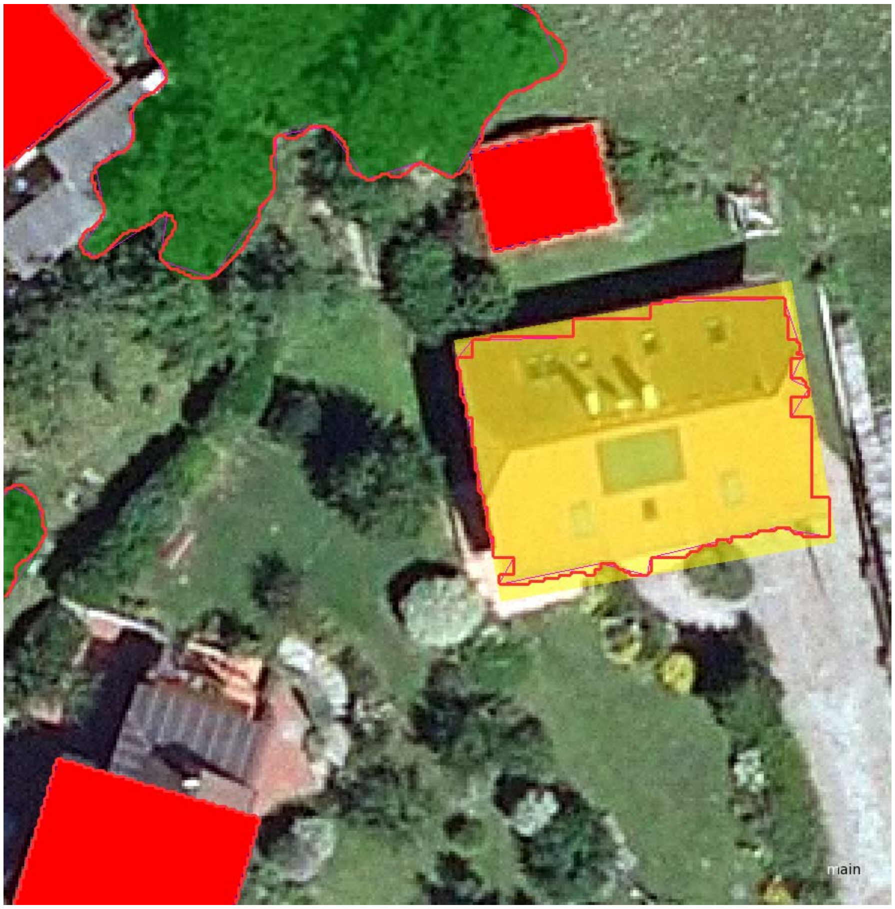
Number of cycles 1

Algorithm Description

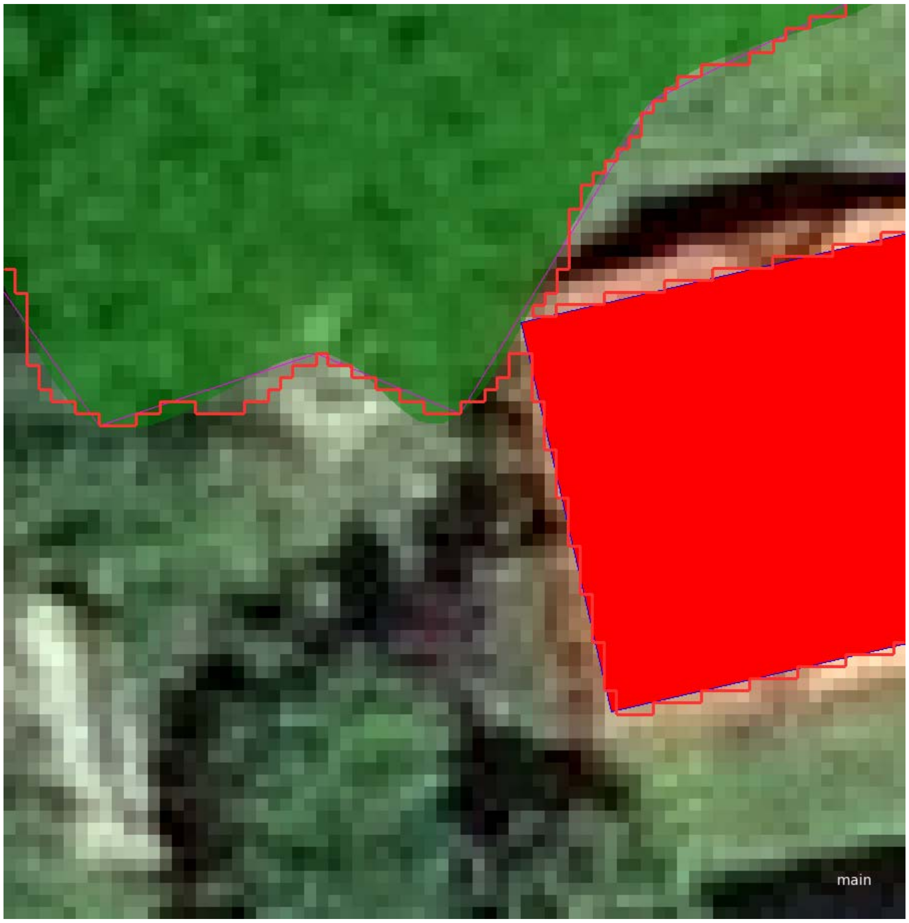
Generalize polygons into rectilinear (orthogonal) polygons at specified granularity.

Algorithm parameters

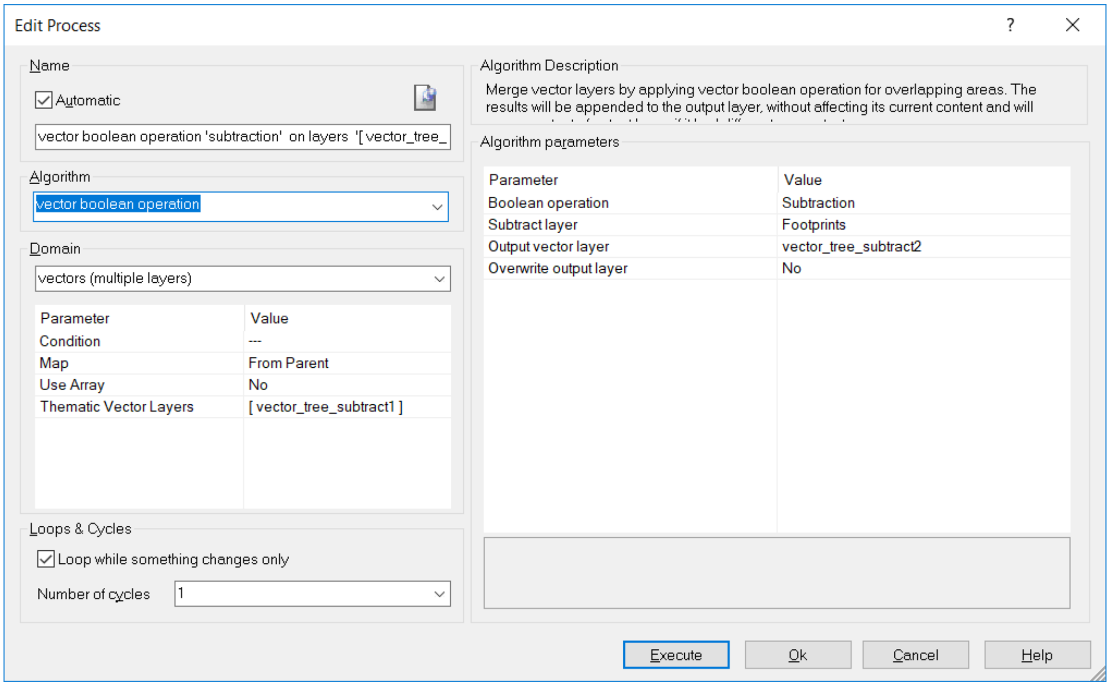
Parameter	Value
Output vector layer	vector_NewBuildings_rectilinear
Chessboard size (pixels)	20
Merge threshold	0.25
Main direction	<auto-detect>



- One more problem to fix = the overlap between the vector objects:



- Let’s subtract one vector layer from another.



☑ Automatic

vector boolean operation 'subtraction' on layers '[vector_tree_

vector boolean operation

vectors (multiple layers)

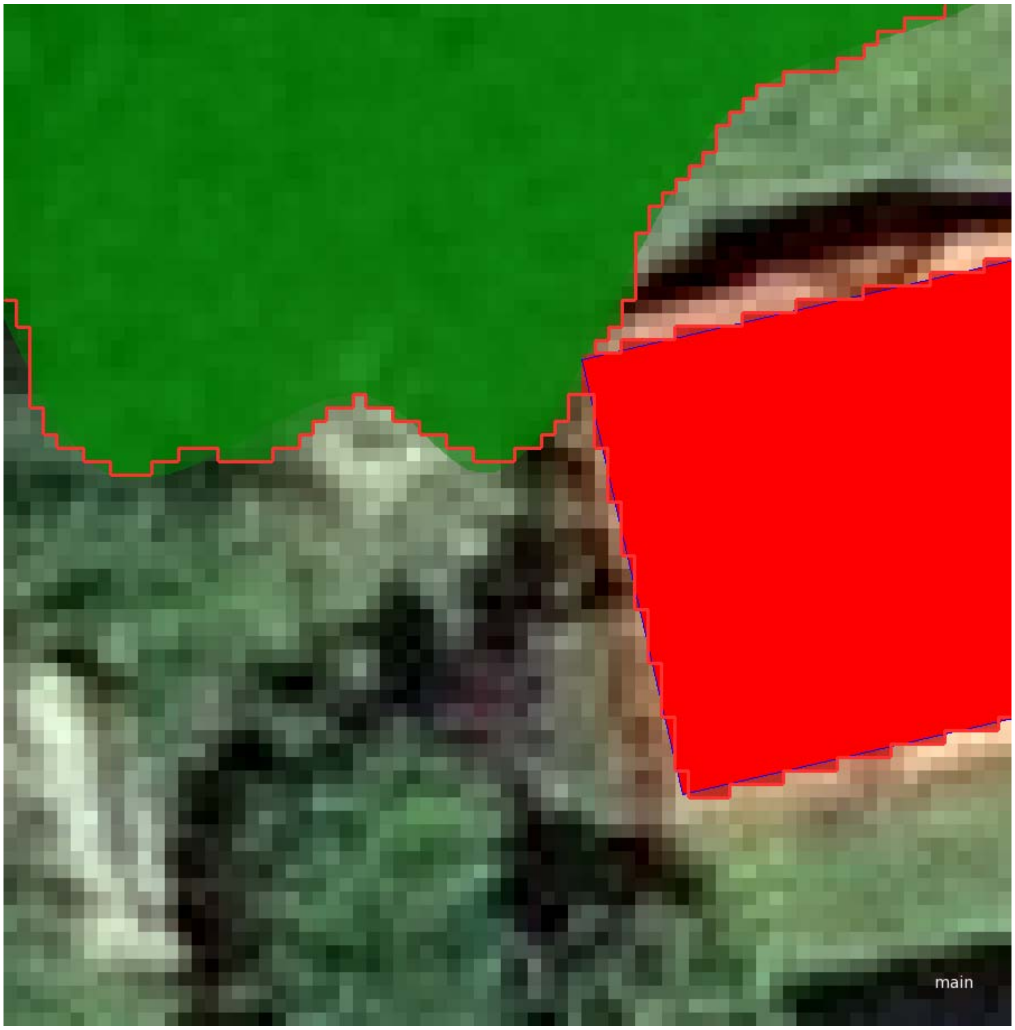
Parameter	Value
Condition	---
Map	From Parent
Use Array	No
Thematic Vector Layers	[vector_tree_subtract1]

☑ Loop while something changes only

Number of cycles 1

Merge vector layers by applying vector boolean operation for overlapping areas. The results will be appended to the output layer, without affecting its current content and will

Parameter	Value
Boolean operation	Subtraction
Subtract layer	Footprints
Output vector layer	vector_tree_subtract2
Overwrite output layer	No



- Vector Integration – snapping or averaging.
- After export to ArcGIS: