eCognition Developer

Tutorial 6 - working with the accuracy assessment tool

www.trimble.com

TRANSFORMING THE WAY THE WORLD WORKS

Trimble.

Introduction

About this Tutorial

This tutorial introduces the user to working with the Accuracy Assessment tool within **eCognition Developer**.

The goal of this tutorial is to give you an introduction to working with sample statistics and the 'Accuracy Assessment' tool to generate a typical **error matrix** that computes the user's and producer's accuracies as well as kappa and other useful statistics related to the accuracy of the classification.

This Module has two lessons:

- Lesson 1 Working with sample statistics
- Lesson 2 Accuracy Assessment

Further information about eCognition products is available on our website:

www.eCognition.com

Requirements

To perform this Guided Tour, you will need:

- **eCognition Developer** installed on a computer
- A computer **mouse** is highly recommended

All steps of this tutorial can be done using the **eCognition Developer** or the free-trial version.

This tutorial is designed for self-study.

Data included with the Tutorial

Image data

We will be working with a 4-band QuickBird multispectral satellite image (*.tif) file in this tutorial:

- 'Image.TIF' contains the **RGB** and **NIR** data

Thematic data

In addition to the image data, we will also use two shapefiles:

- '**Samples_training.shp**' contains the training samples used within the supervised classification.
- '**Samples_validation.shp**' contains the validation samples (e.g. ground truth points) used to perform the accuracy assessment.

Rule Sets

A Rule Sets is available representing the final state of Rule Set development. Whenever the tutorial refers to a Rule Set, it can be found in the tutorial folder.

Project

An eCognition Project is provided for this tutorial and can be found in the tutorial folder.

Recommendations

This tutorial works with the supervised classification functionality of eCognition Developer. If you are unfamiliar with the concept of supervised classification, we suggest reviewing this prior to starting this tutorial. You can find a description of the supervised classification algorithms and their usage in the eCognition Developer **Reference Book** and **User Guide**.

Lesson 1 – Working with sample statistics

1.0 Lesson content

- Introduction to sample statistics
- Importing samples from a shapefile
- Creating sample statistics for supervised classification
- Creating validation samples for accuracy assessment

1.1 Introduction to sample statistics

In **eCognition Developer**, you have the possibility to work with so called '**sample statistics**' when training and applying a supervised classification model. Sample statistics provide the user with a transferable and location -independent method that produce robust and accurate results.

eCognition Developer provides algorithms to save, load and use sample statistics. The sample statistics are saved in a *.csv format.

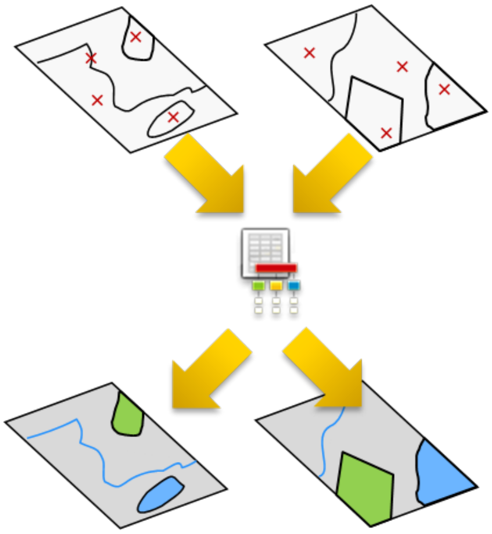


Figure 1: Sample statistics provide a more transferable supervised classification method.

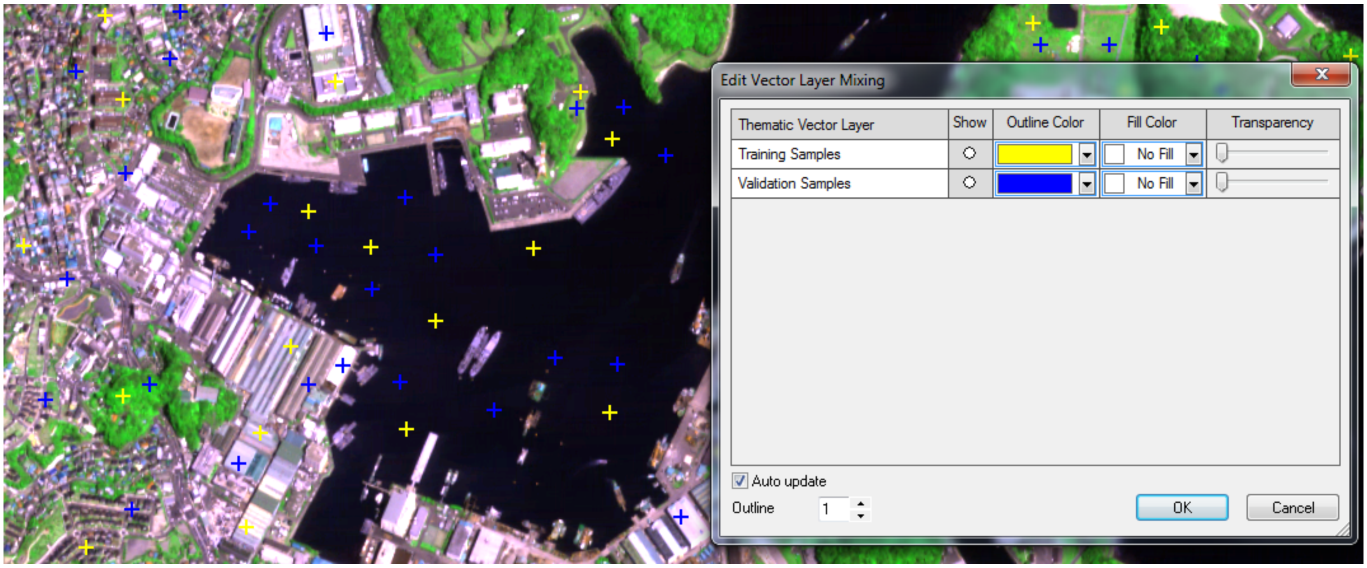
1.2 Importing samples from a shapefile

Samples can be created in several ways with eCognition. In this tutorial, we are working with two sets of sample data stored in separate shapefiles. The information in these sample files, e.g. ground truth data, must be imported into the eCognition Project so it can be used to train the classifier model and validate the accuracy of the resulting classification.

1. Open the eCognition Project file (*.dpr) '**Project.dpr**' in the tutorial folder.
2. Switch to view setting 4 '**Develop Rulesets**' to view the preconfigured Rule Set in the Process Tree window.

If you explore the Project settings within the 'Create/modify Project' dialog, you will see that the Project not only contains a 4-band image, but two shapefiles. Vector files are not automatically displayed in the viewer.

3. Select the 'View/Hide Vector Layers' button from the toolbar to open the '**Vector Layer Mixing**' dialog.
4. Click within the '**Show**' field to display a vector layer.



Edit Vector Layer Mixing

Thematic Vector Layer	Show	Outline Color	Fill Color	Transparency
Training Samples	○	🟨	□ No Fill	
Validation Samples	○	🟦	□ No Fill	

☑ Auto update

Outline 1

Figure 2: Vector layers selected for display and visible within the viewer.

Before any samples can be imported, image objects must be created.

5. **Expand** the Rule Set and expand the section ‘**create objects**’.
6. **Execute** the multiresolution segmentation algorithm to generate image objects. This also creates an image object level called ‘**Main Level**’.

NOTE:
Sample objects are stored in the image object level. If you delete an image object level that contains samples, you will receive a warning. Once the image object level is delete, the sample stored in the level will be gone.

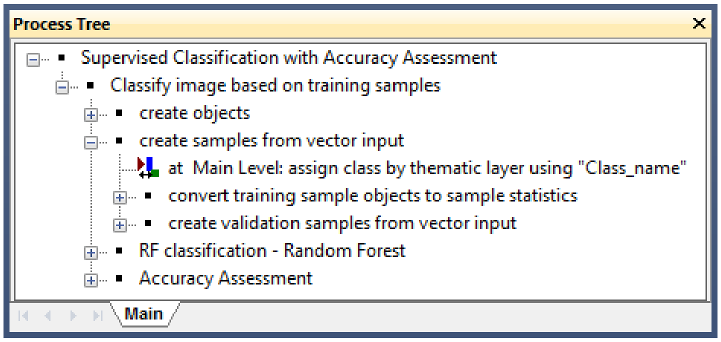
1.2.1 Importing training samples

Importing samples from a shapefile is a simple three-step process:

- Create image objects via a segmentation algorithm
- Classify image objects that correspond (i.e. overlap) with a sample object
- Convert the classified image objects to sample objects

The first set of samples to import will be the training samples. These samples eventually be used to train the classifier model.

1. **Expand** the ‘**create samples from vector input**’ section of the Rule Set. This portion of the Rule Set contains all algorithms needed to import the various sample datasets.

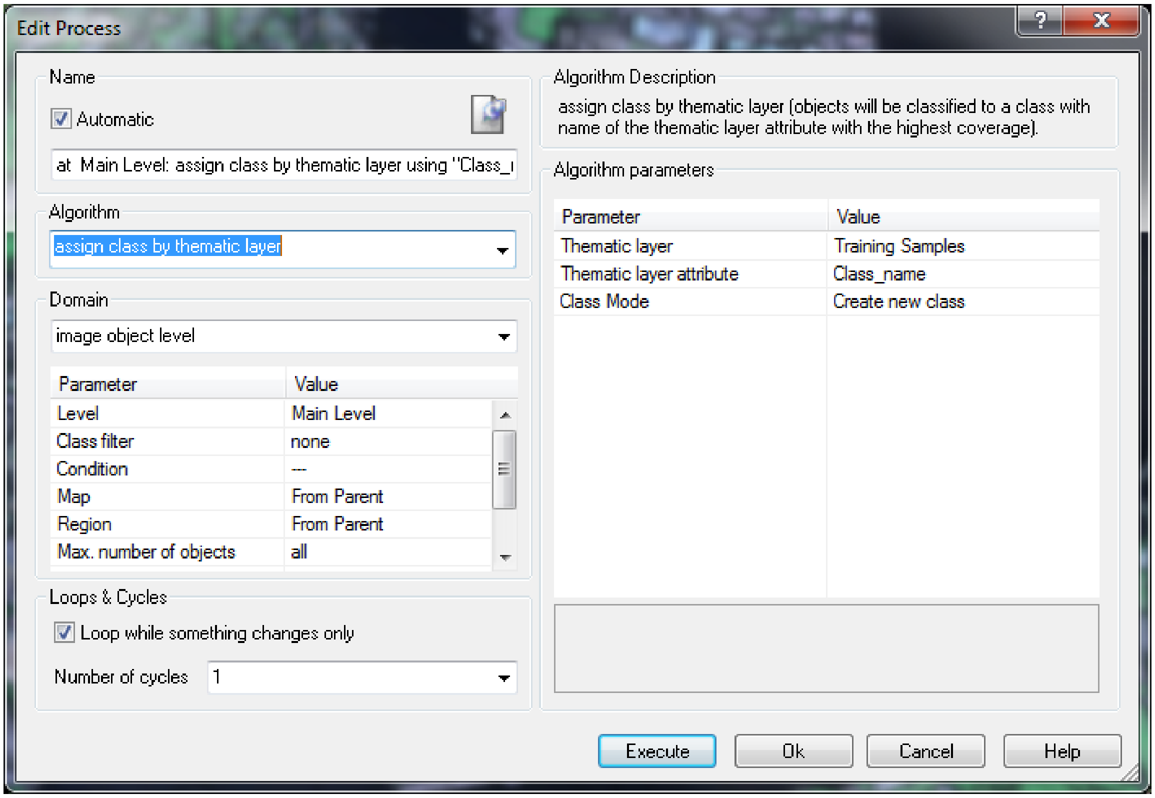


Process Tree

```
⊟ Supervised Classification with Accuracy Assessment
    ⊟ Classify image based on training samples
        ⊞ create objects
        ⊟ create samples from vector input
              at  Main Level: assign class by thematic layer using "Class_name"
            ⊞ convert training sample objects to sample statistics
            ⊞ create validation samples from vector input
        ⊞ RF classification - Random Forest
        ⊞ Accuracy Assessment
```

Figure 3: Process Tree with ‘create samples from vector input’ section expanded.

2. Open the algorithm ‘**at Main Level: assign class by thematic layer using "Class_name"**’. This is an ‘Assign class by thematic layer’ algorithm. It is used to assign image objects to a class based on their relationship to a thematic layer, in this case a shapefile.
    - Within the Domain section, the ‘**Main Level**’ has been selected - this algorithm will only consider image objects in this image object level.
    - In the ‘Thematic layer’ field, the ‘**Training Samples**’ shapefile is defined - image objects that correspond to points within this shapefile will be classified.
    - In the field ‘Thematic layer attribute’, the ‘**Class_name**’ attribute column has been selected - objects will be classified based on this attribute.
    - Since a class hierarchy may not yet exist, the ‘Class Mode’ field is set to ‘**Create new class**’ - this will generate classes in the class hierarchy that match the naming from the shapefile.



Name

☑ Automatic

at Main Level: assign class by thematic layer using "Class_

Algorithm

assign class by thematic layer

Domain

image object level

Parameter	Value
Level	Main 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

assign class by thematic layer (objects will be classified to a class with name of the thematic layer attribute with the highest coverage).

Algorithm parameters

Parameter	Value
Thematic layer	Training Samples
Thematic layer attribute	Class_name
Class Mode	Create new class

Figure 4: The process to classify image objects based on the corresponding thematic object classification.

3. Execute the algorithm and review the results.

Image objects that overlap with the ‘Training Sample’ vector objects have been classified based on the vector objects class defined in the attribute ‘Class_name’. In addition, classes were automatically generated within the Class Hierarchy window that match the naming within the shapefile.

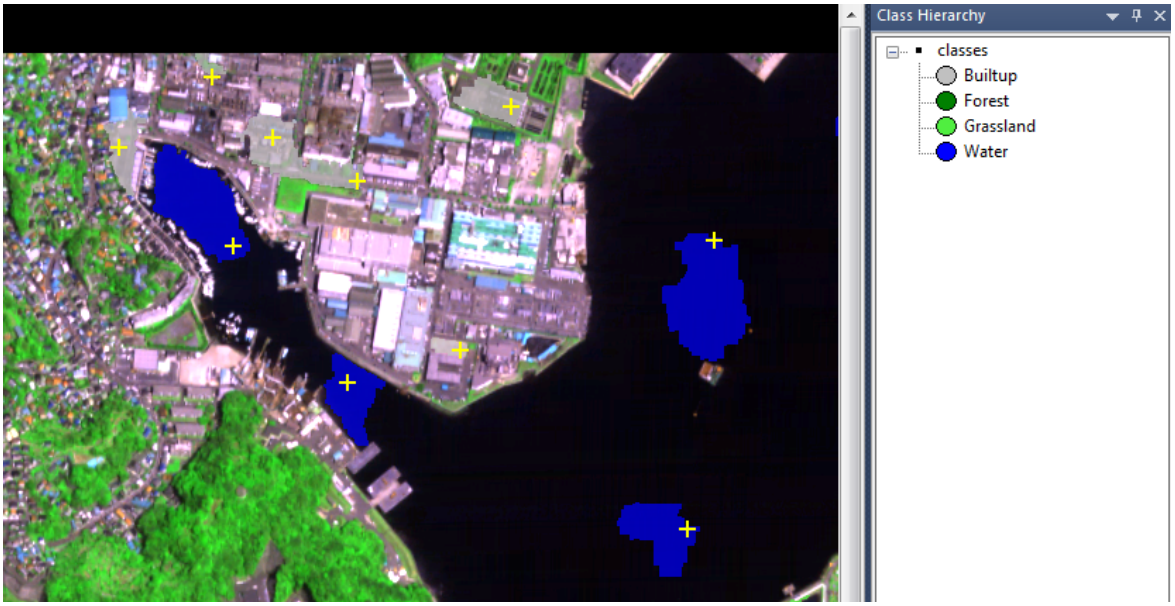


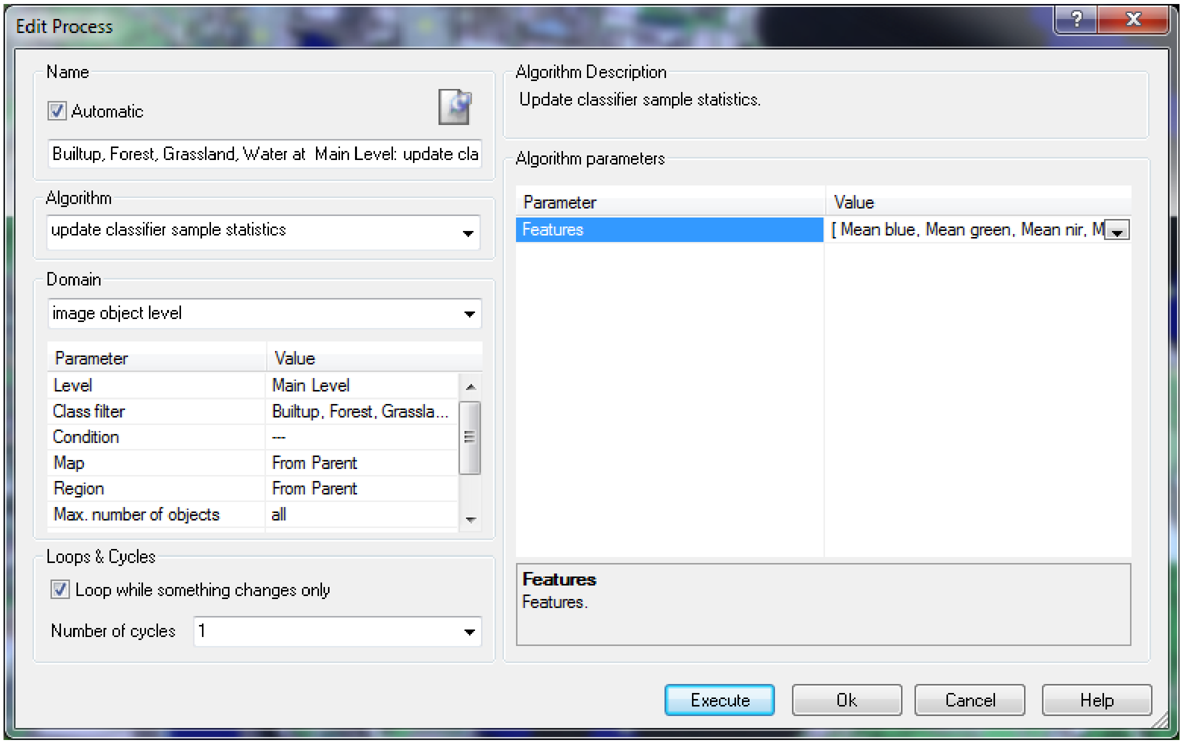
Figure 5: Classification results - classified image objects correspond to overlapping vector object from the 'Training Samples' layer.

Next, we must convert the Training Sample objects to samples. For this purpose, we will store the information within these objects (i.e. feature information) as sample statistics.

4. **Expand** the '**convert training samples object to sample statistics**' section of the Rule Set.

This portion of the Rule Set contains several algorithms that will be used to generate a sample statistic *.csv file that stores the feature information.

5. **Execute** the process '**update classifier sample statistics : clear all**' - this will merely delete any content within the existing sample statistic file if it was run previously.
6. **Open** the second '**update classifier sample statistics**' process. This process will generate sample statistics.
    - The 'Level' field is set to '**Main Level**' as this is the image object level that contains the objects that will be converted to samples.
    - In the 'Class filter' field, classes are defined for the objects to be converted. In this case all four classes are selected: '**Builtup**', '**Forest**', '**Grassland**' and '**Water**'.
    - In the 'Parameters' section, the '**Features**' field is used to define what features should be stored in the sample statistics file.
      These are the features that the supervised classification model will use to calculate the class description. As you can see, some features have already been selected



Edit Process

Name

☑ Automatic

Builtup, Forest, Grassland, Water at Main Level: update cla

Algorithm

update classifier sample statistics

Domain

image object level

Parameter	Value
Level	Main Level
Class filter	Builtup, Forest, Grassla...
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

Update classifier sample statistics.

Algorithm parameters

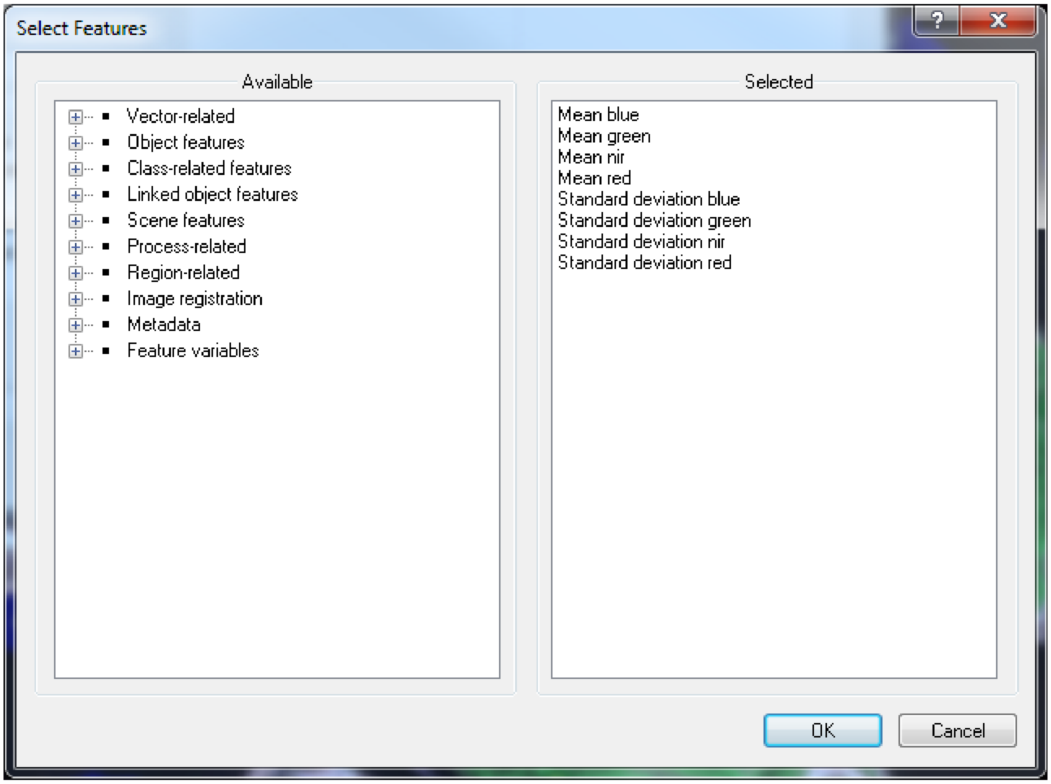
Parameter	Value
Features	[Mean blue, Mean green, Mean nir, M

Features

Features.

Figure 6: Process settings to create sample statistics.

7. Click within the '**Features**' field to access the drop-down menu, select the '**Select features…**' option. This will open the 'Select Features' dialog.
   Selected features are listed on the right and available features are listed on the left.



Select Features

Available

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

Selected

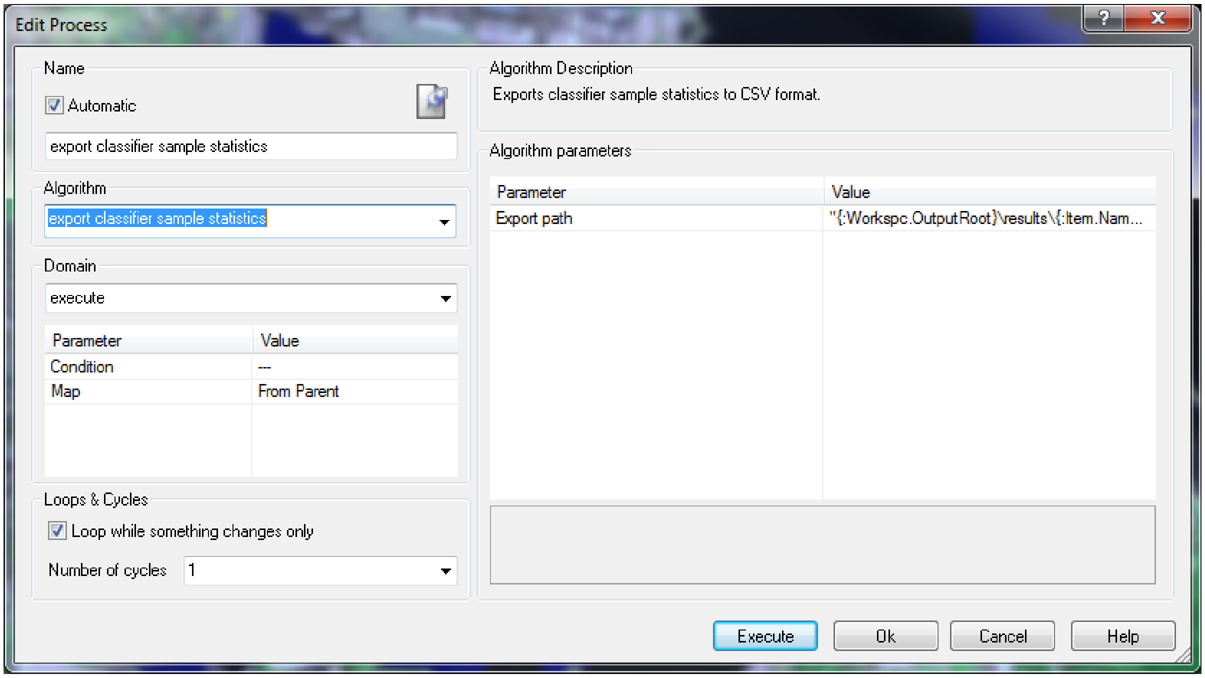
Mean blue
Mean green
Mean nir
Mean red
Standard deviation blue
Standard deviation green
Standard deviation nir
Standard deviation red

Figure 7: The 'Select Features' dialog.

8. Click ‘**Cancel**’ to return to the process.
9. **Execute** the process.

The sample statistics are stored as a variable and are not immediately visible to the user. In addition, they are not yet available for use in other Projects.

10. **Open** the ‘**export classifier sample statistics**’ algorithm. This is used to save the sample statistics to an external *.csv file
    - The ‘Export path’ field is used to define **where the file will be saved**. If you leave the default settings, the file will be saved within a ‘results’ folder where the initial image file is stored.



Name

☑ Automatic

export classifier sample statistics

Algorithm

export classifier sample statistics

Domain

execute

Parameter	Value
Condition	---
Map	From Parent

Loops & Cycles

☑ Loop while something changes only

Number of cycles 1

Algorithm Description

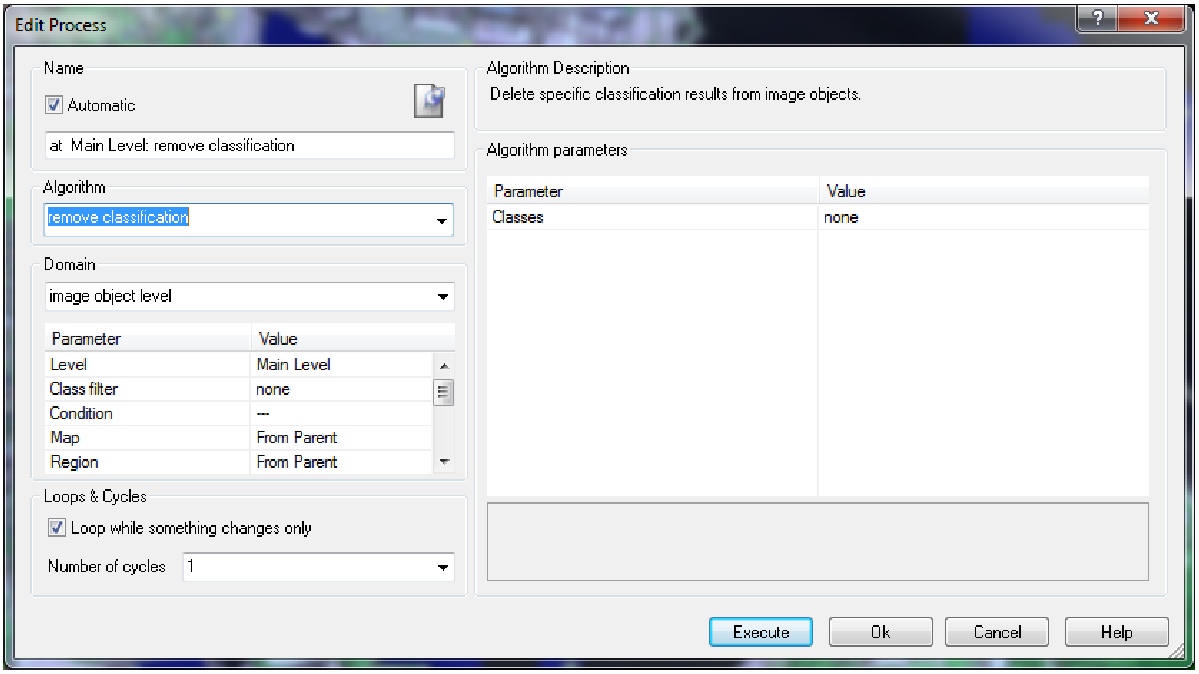
Exports classifier sample statistics to CSV format.

Algorithm parameters

Parameter	Value
Export path	"{:Workspc.OutputRoot}\results\{:Item.Nam...

Figure 8: Process settings to export the sample statistics.

11. **Execute** the process and check your tutorial folder to see if now contains a ‘**results**’ folder.
12. Finally, the current classification is deleted. **Open** the algorithm ‘**at Main Level: remove classification**’ - this is a ‘**remove classification**’ algorithm.
    - All the default settings have been maintained - all classifications will be removed, i.e. all classified image objects will return to the default class ‘unclassified’



Name

☑ Automatic

at Main Level: remove classification

Algorithm: remove classification

Domain: image object level

Parameter	Value
Level	Main Level
Class filter	none
Condition	---
Map	From Parent
Region	From Parent

Loops & Cycles

☑ Loop while something changes only

Number of cycles: 1

Algorithm Description

Delete specific classification results from image objects.

Algorithm parameters

Parameter	Value
Classes	none

Figure 9: Process settings to remove existing classification.

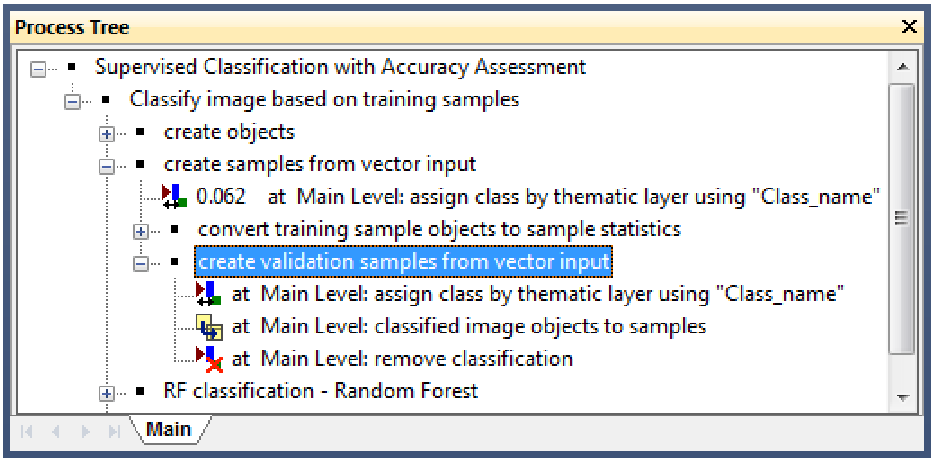
The initial classification was simply used to convert the Training Sample objects into samples. Since the sample information is now stored in a sample statistic file, the classification is no longer needed.

We will come back to these samples later in the Rule Set.

1.2.2 Importing validation samples

Next, the validation samples will be generated. Rather than converting these to sample statistics, these will be simply converted to **sample objects** and stored within the image object level 'Main Level'.

1. **Expand** the '**create validation samples from vector input**' section of the Rule Set. It contains three algorithms that correspond to the simple three step process discussed earlier.

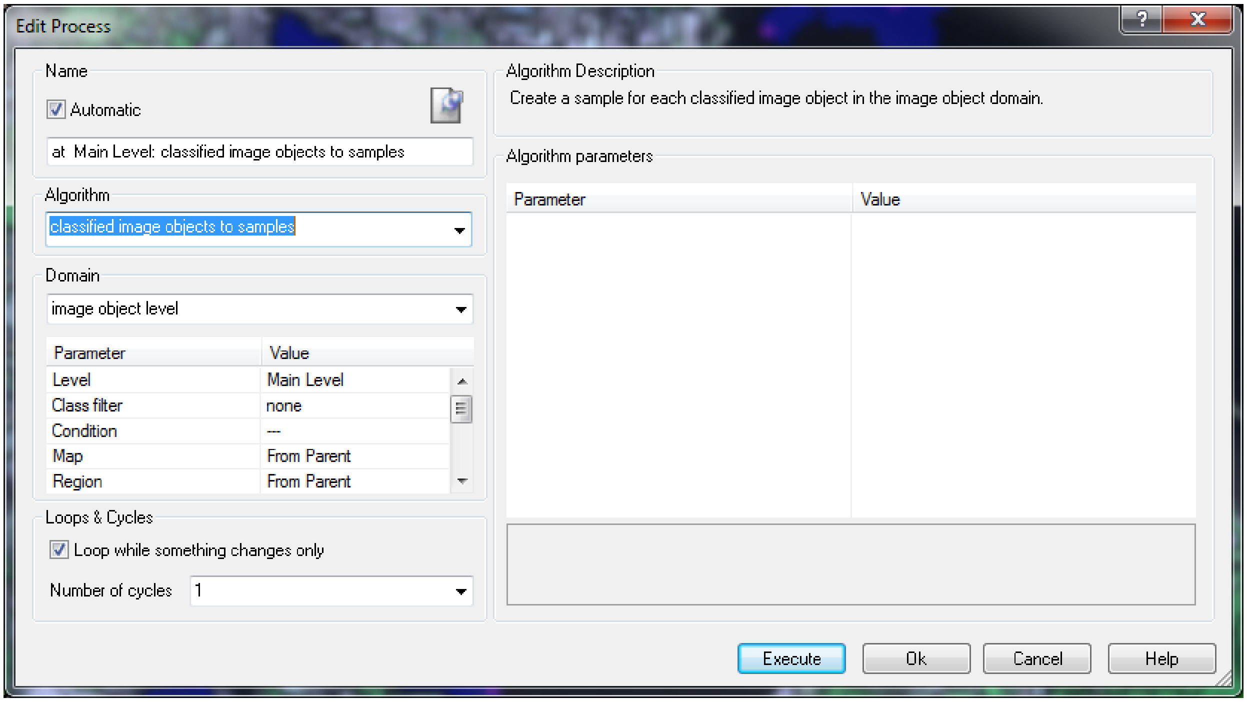


```
Supervised Classification with Accuracy Assessment
    Classify image based on training samples
        create objects
        create samples from vector input
            0.062  at  Main Level: assign class by thematic layer using "Class_name"
            convert training sample objects to sample statistics
            create validation samples from vector input
                at  Main Level: assign class by thematic layer using "Class_name"
                at  Main Level: classified image objects to samples
                at  Main Level: remove classification
        RF classification - Random Forest
```

Figure 10: Process Tree showing the three algorithms that will be used to convert the vector objects to sample objects.

2. Again, the first step is to classify image objects that correspond to vector objects in the shapefile 'Validation Samples'.
   Execute the '**assign class by thematic layer**' algorithm and examine the results.

3. **Open** the '**classified image objects to samples**' algorithm.
   - The default settings have been kept. This will convert all classified image objects to sample objects.



Edit Process

Name

☑ Automatic

at Main Level: classified image objects to samples

Algorithm

classified image objects to samples

Domain

image object level

Parameter	Value
Level	Main Level
Class filter	none
Condition	---
Map	From Parent
Region	From Parent

Loops & Cycles

☑ Loop while something changes only

Number of cycles 1

Algorithm Description

Create a sample for each classified image object in the image object domain.

Algorithm parameters

Parameter	Value

Figure 11: Process settings to convert classified image objects to sample objects.

4. Again, the classification is removed now that the sample objects have been created the classification no longer has a purpose.
   Execute the algorithm.
5. Check the status of the sample objects - click the 'View Samples' button in the 'View Settings' toolbar to only display sample objects.

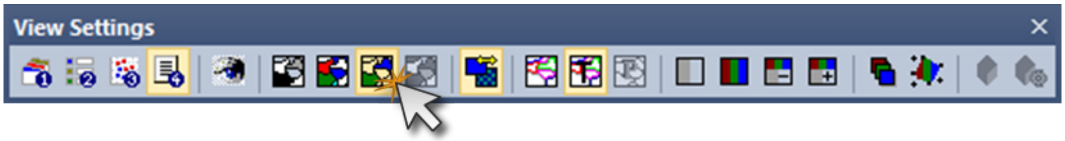


Figure 12: 'View Settings' toolbar with 'View Samples' setting selected.

The classified image objects are gone, but the sample objects exist within the image object level.

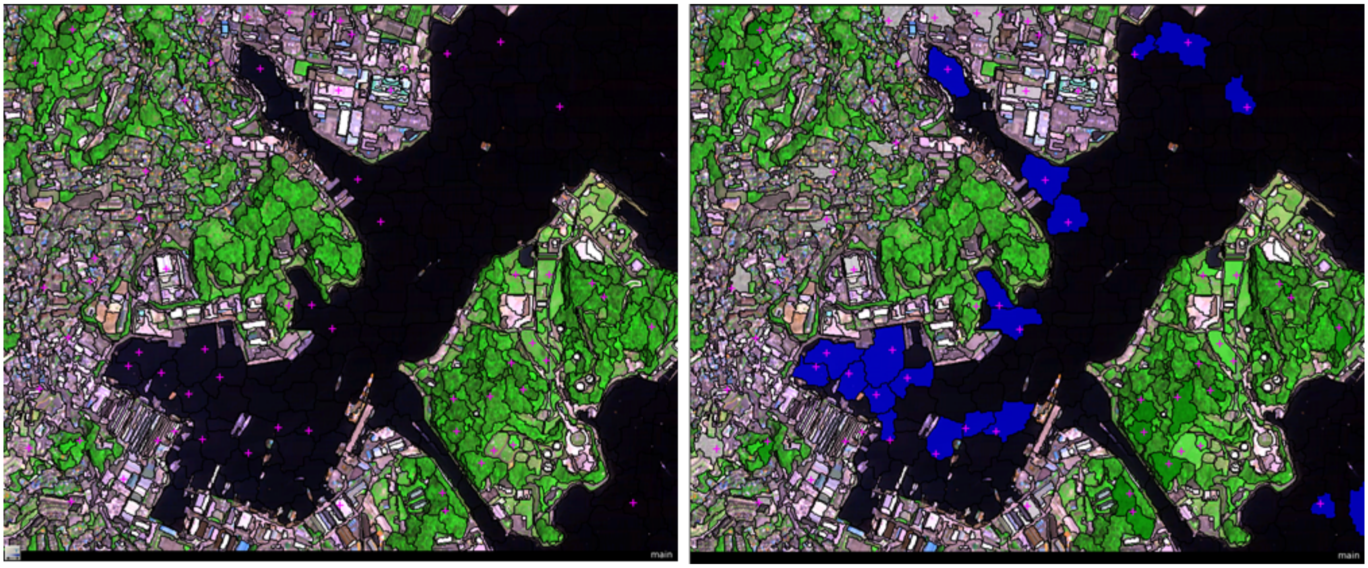


Figure 13: Classification view (left); Sample view (right).

1.3 Train and apply the supervised classification model

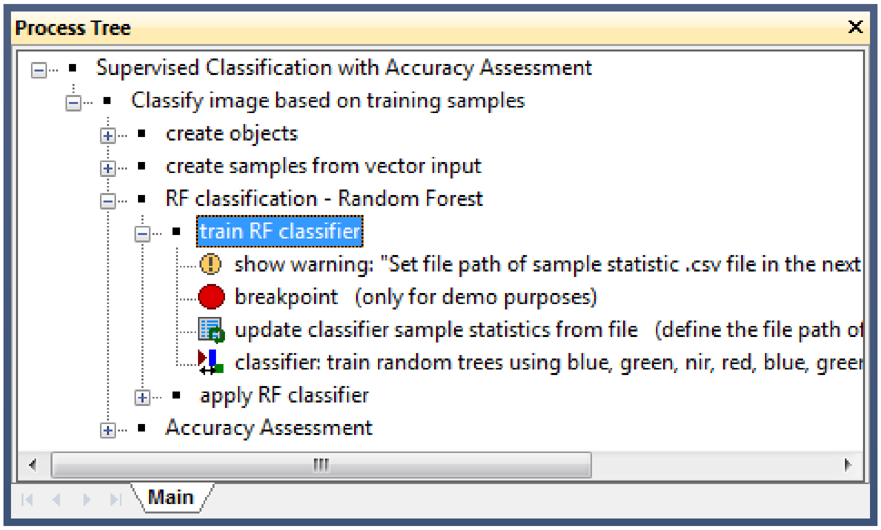
This tutorial will not go into detail regarding the various supervised classification methods. Here we will demonstrate how to train and apply a supervised classification model based on sample statistics.

In **Chapter 1.2.1** of this tutorial, a sample statistics file was exported that contains the feature information that will be used to train a supervised classifier model. This *.csv file will be used in this section of the Rule Set within the ‘**classifier**’ algorithm.

1.3.1 Training the classifier

Before a supervised classification can be applied, the model on which it is based must be trained.

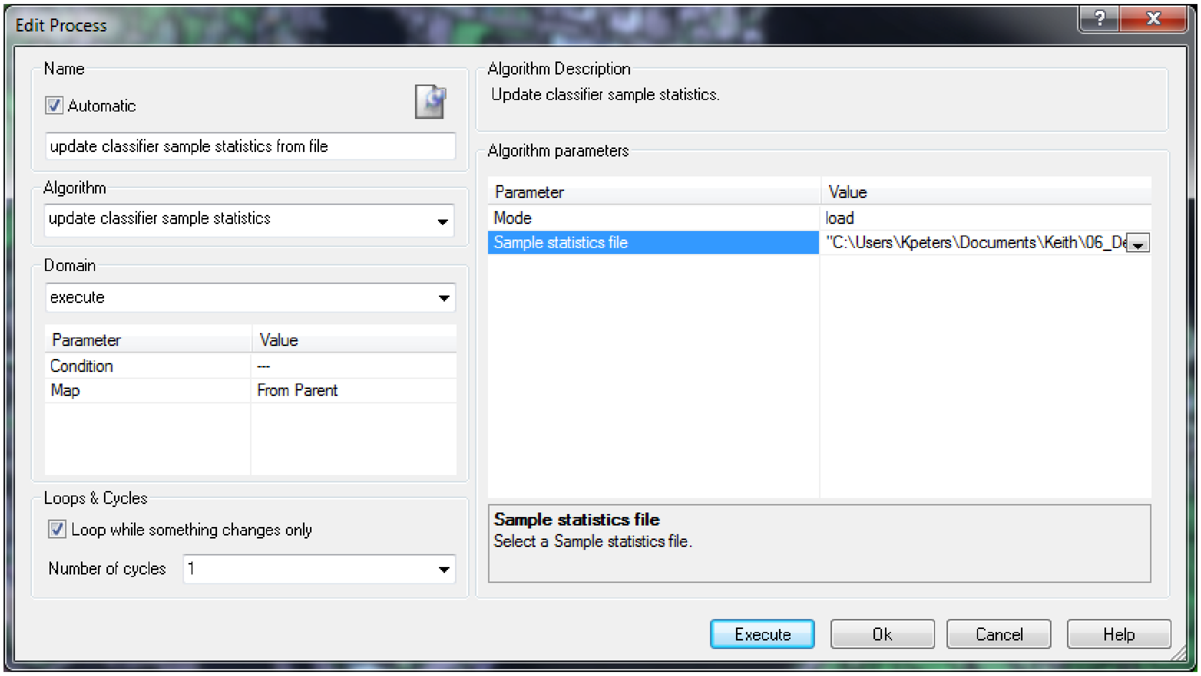
1. **Expand** the ‘RF classification - Random Forest’ section of the Rule Set as well as the child process ‘**train RF classifier**’.



- Supervised Classification with Accuracy Assessment
  - Classify image based on training samples
    - create objects
    - create samples from vector input
    - RF classification - Random Forest
      - train RF classifier
        - show warning: "Set file path of sample statistic .csv file in the next
        - breakpoint (only for demo purposes)
        - update classifier sample statistics from file (define the file path of
        - classifier: train random trees using blue, green, nir, red, blue, greer
      - apply RF classifier
    - Accuracy Assessment

Figure 14: Process Tree with Rule Set section for training the classifier selected.

2. There are several algorithms included here for **demonstration** purposes:
   - The '**show warning**' algorithm will remind the user to define the file path for the sample statistics file as this is different on your machine.
   - The **breakpoint** simply stops Rule Set processing at this point to give the user a chance to adjust the file path. Once you have made the needed changes you can continue processing.
3. **Open** the '**update classifier sample statistics**' algorithm.
   - The 'Mode' is set to '**load**'
   - Within the 'Sample statistics file' field, use the **drop-down** menu and select '**Browse…**'. Then select the ***.csv** file from the '**results**' folder and confirm with '**OK**'.



Name

☑ Automatic

update classifier sample statistics from file

Algorithm: update classifier sample statistics

Domain: execute

Parameter	Value
Condition	---
Map	From Parent

Loops & Cycles

☑ Loop while something changes only

Number of cycles: 1

Algorithm Description: Update classifier sample statistics.

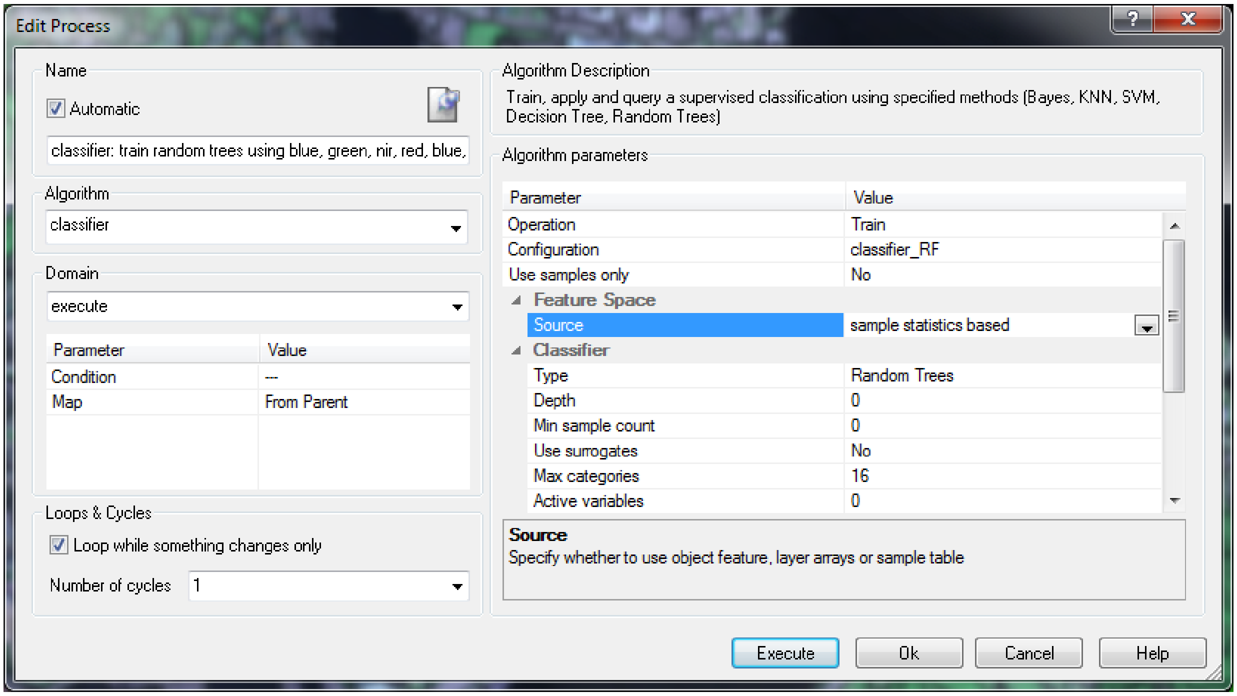
Algorithm parameters

Parameter	Value
Mode	load
Sample statistics file	"C:\Users\Kpeters\Documents\Keith\06_De

Sample statistics file
Select a Sample statistics file.

Figure 15: Process settings to load the sample statistics file.

4. **Execute** the process.
5. **Open** the '**classifier**' algorithm and examine the parameters.
   - The 'Operation' is set to '**Train**'
   - In the 'Configuration' field, the name of the variable that will store the classier model is defined - the use can type any name they like here. In this tutorial, the model is called '**classifier_RF**'.
     It is possible to train multiple models to account for different classifier types.
   - The 'Use samples only' field is set to '**No**' since we do not want to use any **object samples** to train the classifier.
   - The 'Source' of the feature space is set to '**sample statistics based**'
   - In this example, a '**Random Trees**' (i.e. Random Forest) classifier 'Type' will be trained based on the default settings.


Edit Process

Name

☑ Automatic

classifier: train random trees using blue, green, nir, red, blue,

Algorithm: classifier

Domain: execute

Parameter	Value
Condition	---
Map	From Parent

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	classifier_RF
Use samples only	No
Feature Space	
Source	sample statistics based
Classifier	
Type	Random Trees
Depth	0
Min sample count	0
Use surrogates	No
Max categories	16
Active variables	0

Source
Specify whether to use object feature, layer arrays or sample table

Figure 16: Process settings to train the classifier based on sample statistics.

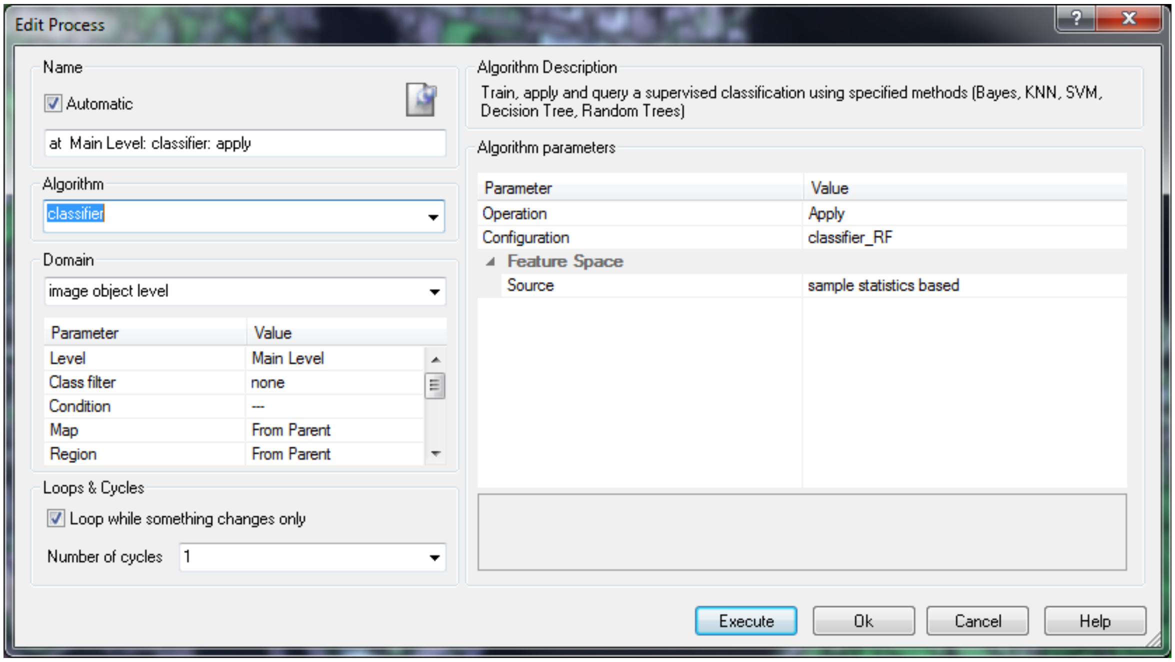
6. **Execute** the algorithm to train the classifier.

1.3.2 Apply the classifier

It may appear as if nothing has happened within the viewer. This is because we have simply trained the classifier model.

Now we will apply the Random Trees model to classify the image objects.

1. **Expand** the '**apply RF classifier**' section and **open** the '**classifier**' algorithm.
   - The Domain is set to '**image object level**' and the level is defined as '**Main Level**'
   - The 'Operation' is now set to '**Apply**'
   - The 'Configuration' containing the trained model is defined as '**classifier_RF**'
   - The feature space 'Source is again set to '**sample statistics based**'



Edit Process

Name

☑ Automatic

at Main Level: classifier: apply

Algorithm

classifier

Domain

image object level

Parameter	Value
Level	Main Level
Class filter	none
Condition	---
Map	From Parent
Region	From Parent

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	classifier_RF
Feature Space	
Source	sample statistics based

Figure 17: Process settings to apply the classifier based on sample statistics.

2. Execute the algorithm and examine the results.

The image has now been classified based on the classifier model and training sample statistics.

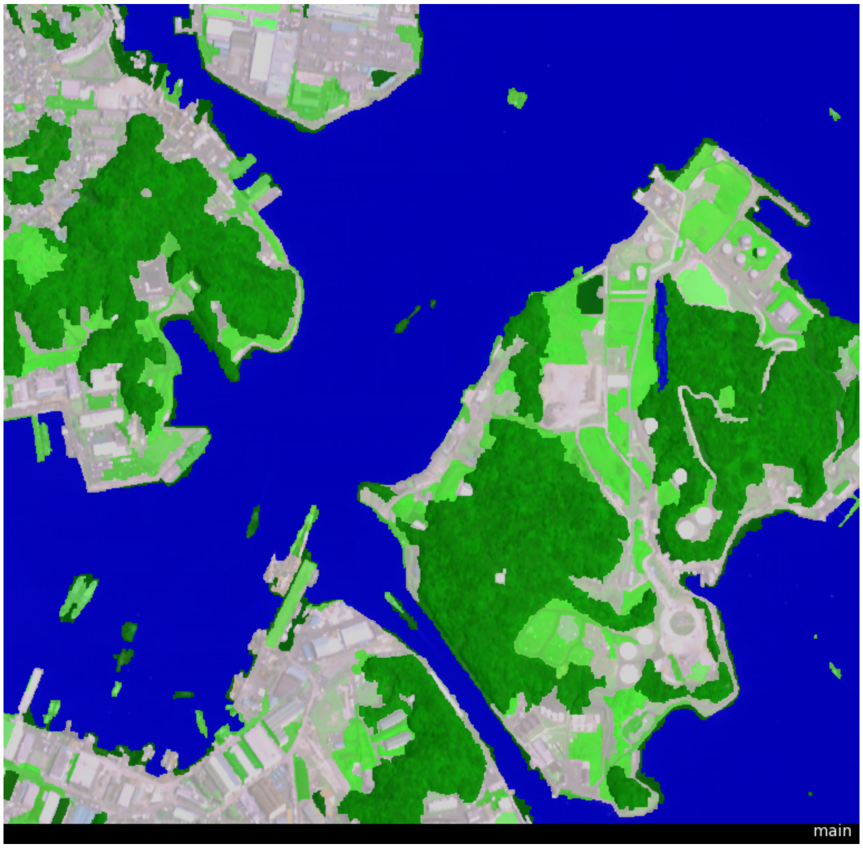


Figure 18: Classification results.

Lesson 2 - Performing an accuracy assessment

2.0 Lesson content

- Generating an error matrix based on samples
- Exporting the error matrix results

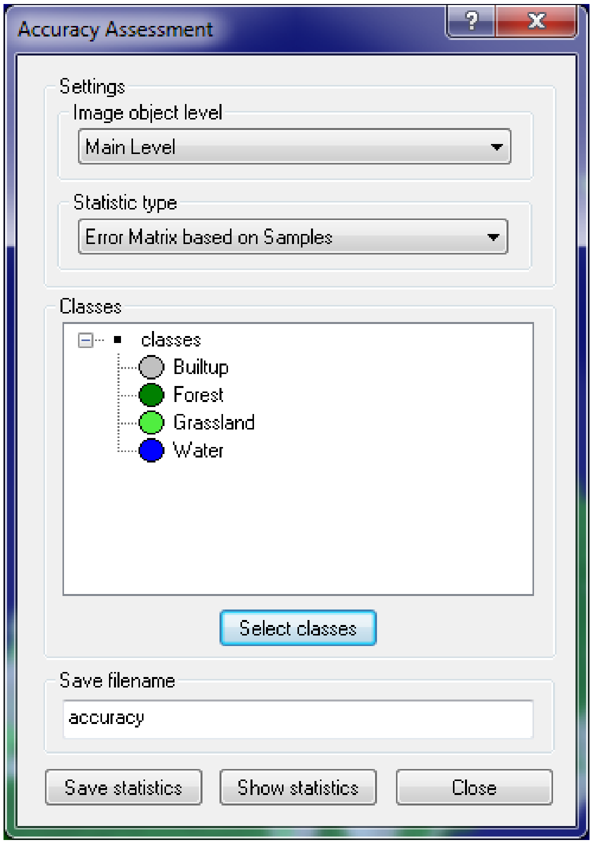
Now that the classification is complete, it is common to examine the accuracy of the results. **eCognition Developer** provides users with '**Accuracy Assessment**' tool for this purpose.

In this tutorial, we will use the Accuracy Assessment tool to generate an error matrix based on the validation samples created in **Chapter 1.2.2**.

2.1 Calculate error matrix based on samples

The Accuracy Assessment tool is available within the **Tools** menu.

1. **Open** the Accuracy Assessment tool dialog - **Tools** > **Accuracy Assessment**
2. Make sure the image object level containing the validation samples is selected from the '**Image object level**' drop-down menu.
   It should be set to '**Main Level**'.
3. **Select** the '**Error Matrix based on Samples**' option from the 'Statistics type drop-down menu.
4. Select the classes to calculate the accuracy statistics for by clicking '**Select classes**' - the 'Select Classes for Statistics' dialog opens.
   Click on a class in the available window to select. All **four** classes should be selected. Confirm with '**OK**'



Accuracy Assessment

Settings

Image object level: Main Level

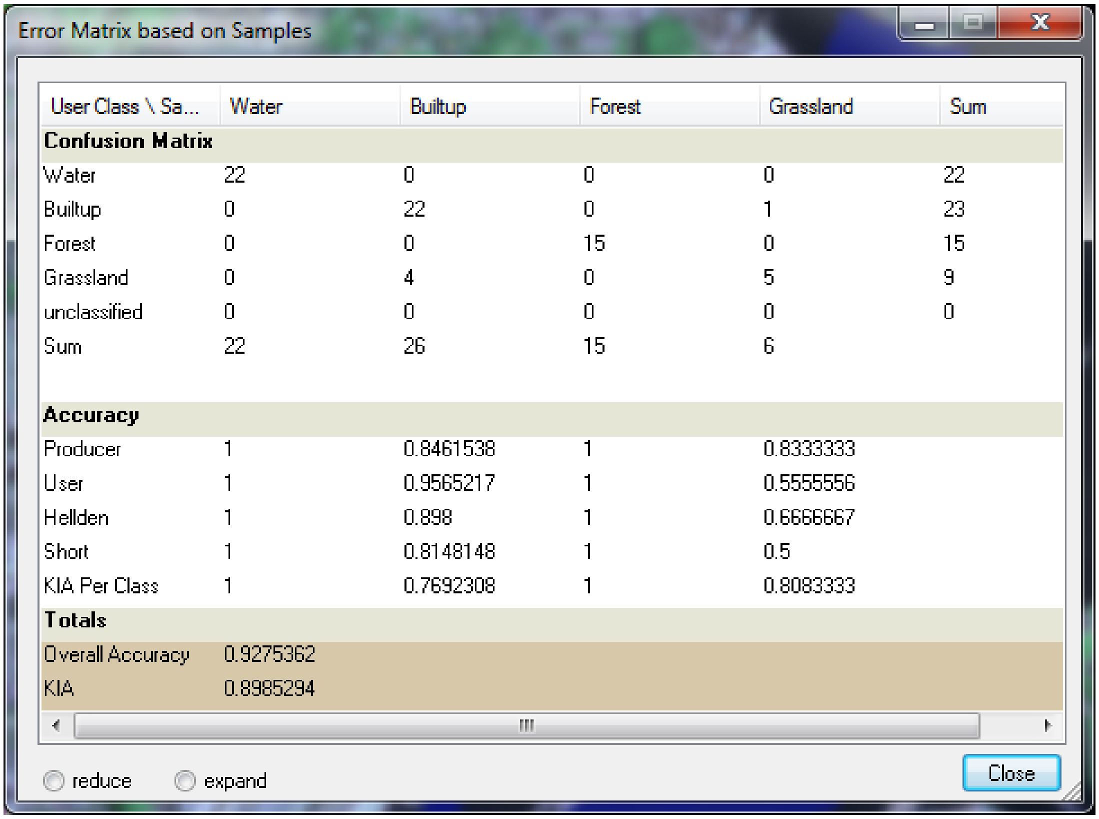
Statistic type: Error Matrix based on Samples

Classes:
- classes
  - Builtup
  - Forest
  - Grassland
  - Water

Save filename: accuracy

Figure 19: The Accuracy Assessment dialog.

5. Click the ‘**Show statistics**’ button to view the **error matrix**.



User Class \ Sa...	Water	Builtup	Forest	Grassland	Sum
Confusion Matrix					
Water	22	0	0	0	22
Builtup	0	22	0	1	23
Forest	0	0	15	0	15
Grassland	0	4	0	5	9
unclassified	0	0	0	0	0
Sum	22	26	15	6	
Accuracy					
Producer	1	0.8461538	1	0.8333333	
User	1	0.9565217	1	0.5555556	
Hellden	1	0.898	1	0.6666667	
Short	1	0.8148148	1	0.5	
KIA Per Class	1	0.7692308	1	0.8083333	
Totals					
Overall Accuracy	0.9275362				
KIA	0.8985294				

Figure 20: Error matrix preview.

2.2 Exporting the error matrix

The error matrix statistics can be saved as a *.csv file.

1. Define the name of the error matrix statistics file in the ‘**Save filename**’ field.
2. Click the ‘**Save statistics**’ button.

The results are saved in the tutorial file or the folder containing the initial input image when working with a Project or in the Workspace root folder if using a Workspace.

3. Click ‘**Close**’ to return to the Project.

.

Where to get additional help & information?

The eCognition Community

The eCognition Community helps to share knowledge and information within the user, partner, academic and developer community to benefit from each other's experience.

The Community contains content such as:

- **Wiki**: collection of eCognition related articles (e.g. Rule Set tips and tricks, strategies, algorithm documentation...).
- **Discussions**: ask questions and get answers.
- **File exchange**: share any type of eCognition related code such as Rule Sets, Action Libraries, plug-ins...
- **Blogs**: read and write insights about what's happening around our industry…

Share your knowledge and questions with other users interested in using and developing image intelligence applications for Earth Sciences at:

http://community.ecognition.com/.

The User Guide & Reference Book

Together with the software a User Guide and a Reference book is installed. You can access them in the Developer interface in the main menu 'Help>eCognition Developer User Guide' or Reference Book.

The Reference Book lists detailed information about algorithms and features, and provides general reference information.

eCognition Training

eCognition Training Services offer a carefully planned curriculum that provides hands-on, real-world exercises. We are dedicated to enhancing customers' image analysis skills and helping these organizations to accomplish their goals.

Our courses are held in our classrooms around the world and on-site in our customer's facilities. We offer regular Open Training courses, where anyone can register and In-Company Training. We also offer Customized Courses to meet a customer's unique image analysis needs, thereby maximizing the training effect.

For more information please see our website or contact us at: eCognition_Training@trimble.com