D3OTP2 Urban mapping

Image Segmentation and SVM-classification of Sentinel-2 MSI data for urban applications

Introduction

In this training session, you will learn how object based image analysis can be applied in image classification tasks over urban areas. You will first use the KTH-SEG image segmentation algorithm developed at KTH (Ban and Jacob 2013) before classifying a Sentinel-2 scene over Tianjin with a Support Vector Machine (SVM) classifier, also implemented in KTH-SEG.

Data

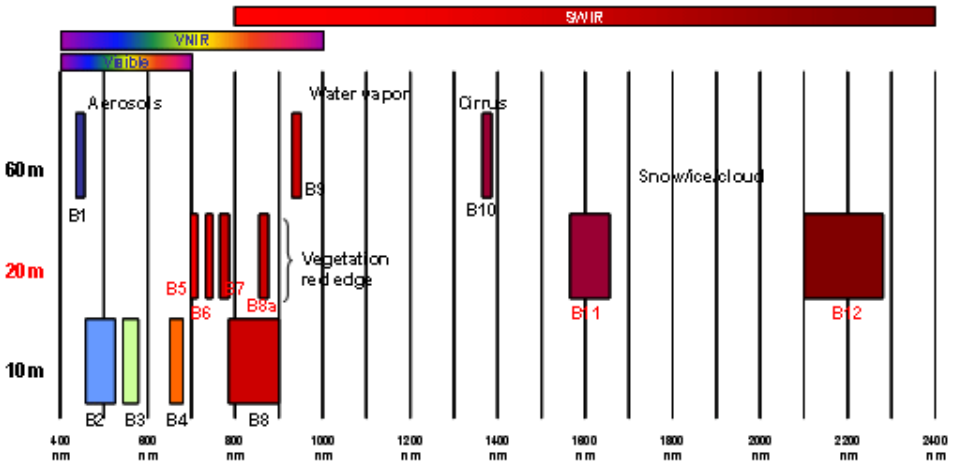
The image that you are going to analyse is a subset of a Sentinel 2 MSI scene over Kunming acquired on 2016-nov-18.

The pair of Sentinel-2 satellites will routinely deliver high-resolution optical images globally, providing enhanced continuity of SPOT- and Landsat-type data. Sentinel-2 carries an optical payload with visible, near infrared and shortwave infrared sensors comprising 13 spectral bands: 4 bands at 10 m, 6 bands at 20 m and 3 bands at 60 m spatial resolution (the latter is dedicated to atmospheric corrections and cloud screening), with a swath width of 290 km https://sentinel.esa.int/web/sentinel/user-guides/sentinel-2-msi/resolutions/radiometric. The 13 spectral bands guarantee consistent time series, showing variability in land surface conditions and minimising any artefacts introduced by atmospheric variability. The mission orbits at a mean altitude of approximately 800km and, with the pair of satellites in operation, has a revisit time of five days at the equator (under cloud-free conditions) and 2–3 days at mid-latitudes. The increased swath width along with the short revisit time allows rapid changes to be monitored, such as vegetation during the growing season.

Data from Sentinel-2 will benefit services associated with, for example, land management by European and national institutes, the agricultural industry and forestry, as well as disaster control and humanitarian relief operations. Imagery for the generation of high-level operational products, such as land-cover maps, land-change detection maps and geophysical variables that use, for example, leaf area index, leaf chlorophyll content and leaf water content will be provided. Images of floods, volcanic eruptions and landslides will also be acquired by Sentinel-2. In essence, Sentinel-2 combines a large swath, frequent revisit, and systematic acquisition of all land surfaces at high-spatial resolution and with a large number of spectral bands, all of which makes a unique mission to serve Copernicus.

In this exercise, four bands at 10 m resolutions in the visible and VNIR spectrum and 6 bands at 20m resolution in the VNIR and SWIR spectrum are used as a subset of the S2-scene. The image was radiometrically resampled to 8-bit and the 10m bands were spatially resampled to 20m resolution for segmentation and classification purposes. The following table summarizes the image bands, their wavelengths and description that are present in the **tianjin.tif** image.

band # in the original image	central wavelength in nm	description	spatial resolution in m	Band # in tianjin.tif
5	705	VNIR	20m	1
6	740	VNIR	20m	2
7	783	VNIR	20m	3
8a	865	VNIR	20m	4
11	1610	SWIR	20m	5
12	2190	SWIR	20m	6
2	490	blue	10m	7
3	560	green	10m	8
4	665	red	10m	9
8	842	NIR	10m	10

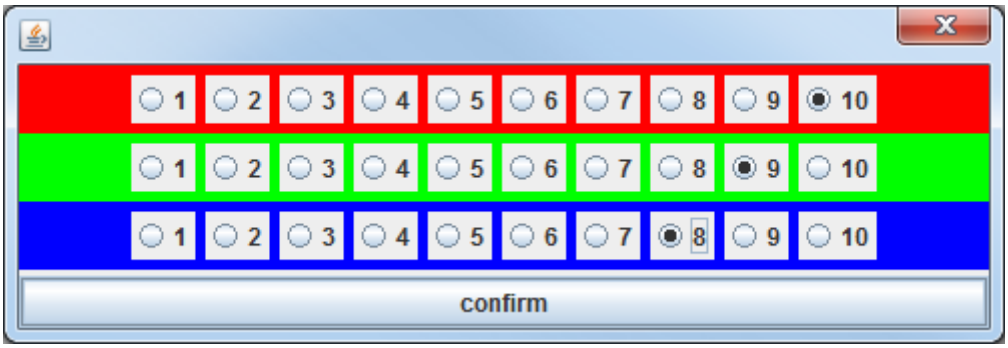


Segmentation

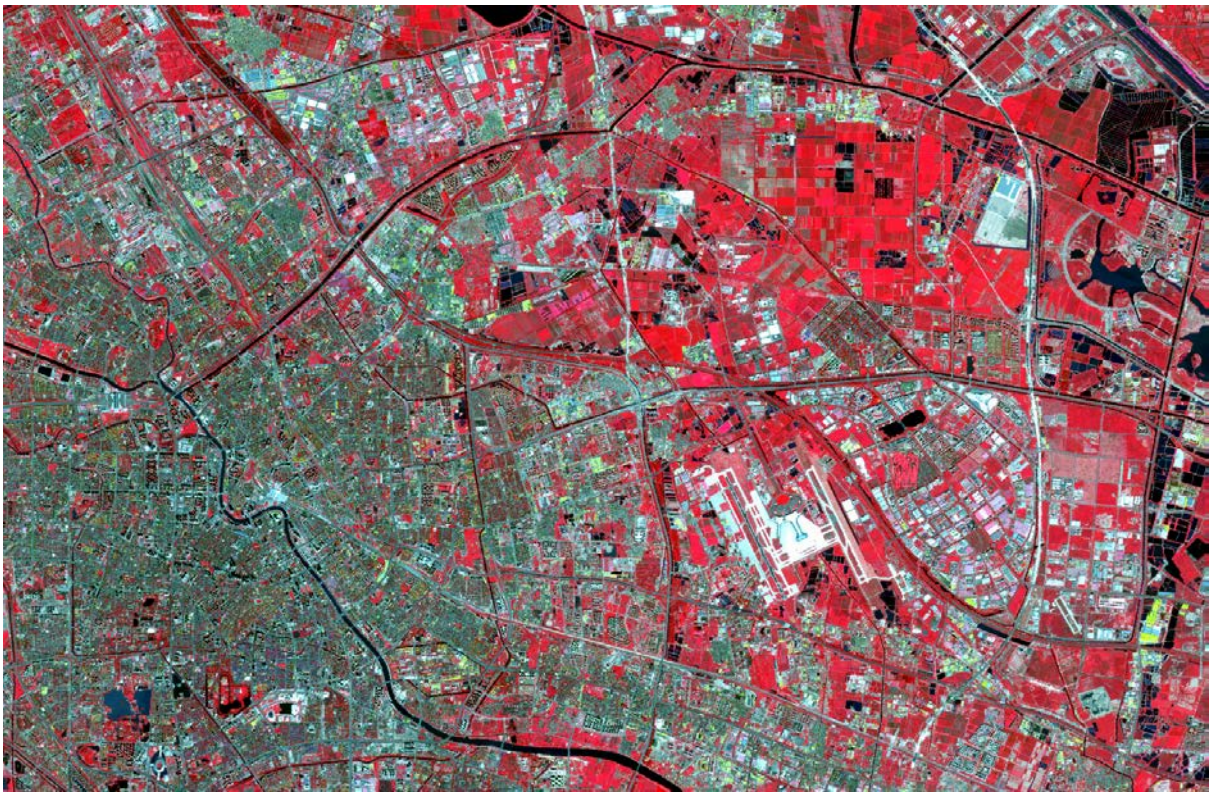
Image segmentation will be performed using the KTH-SEG program. KTH-SEG is an object based image analysis software toolbox which is currently under development at KTH, the Royal Institute of Technology in Stockholm, Sweden. At the time it can work with GeoTIFF as input format for images and as a result shape-files can be exported.

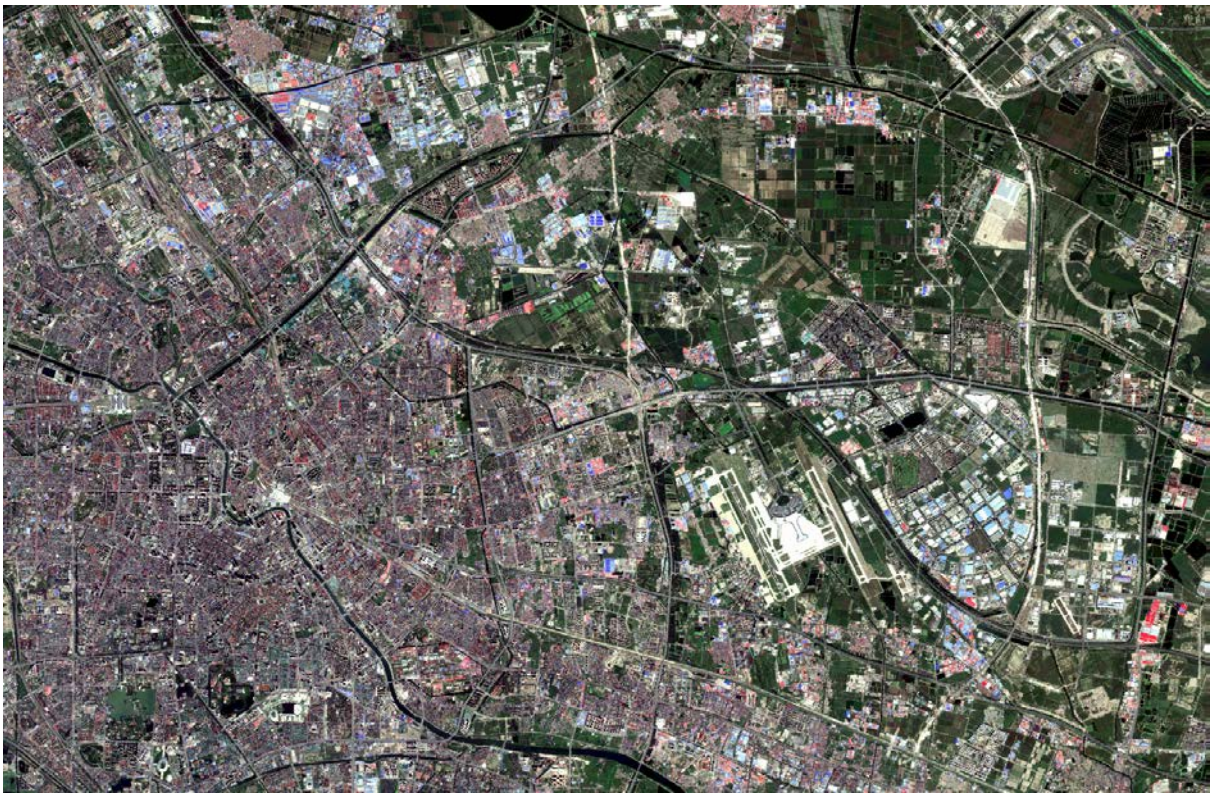
You can launch the software using the **KTH-SEG-run.bat** file. For this to work both the KTH-SEG-run.bat and the KTH-SEG.jar file need to be in the same folder! When executing the .bat file two windows will open: One showing the graphical user interface (GUI) of KTH-SEG and one showing the command line shell. The latter you can use to monitor process during operations such as segmentation. *Please not that a **64-bit** version of Java 8 needs to be installed on the computer for KTH-SEG to run.*

To get started with your analysis you will first need to load an image into KTH-SEG. For this click on the **File** menu and select **open**. Navigate to the location of your *tianjin.tif*. After having selected the image a band selection dialog will appear:

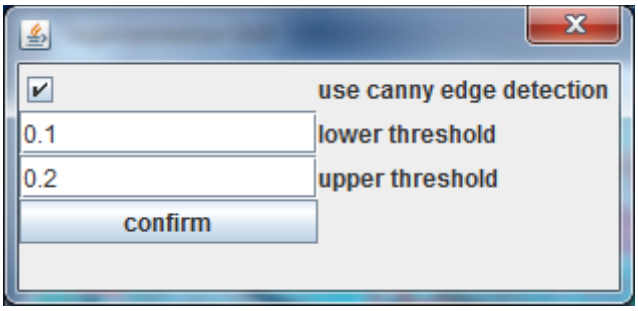


Here you can select which bands are to be displayed in what colour. You can select one colour for each band. Assign a band for each display channel (e.g. Band10: R, Band9: G, and Band8: B) to create a false colour composite. The image will be displayed together with the classified segments in a later step. The following two images show a false-colour-composite and a true-colour-composite composite of the scene we will classify:



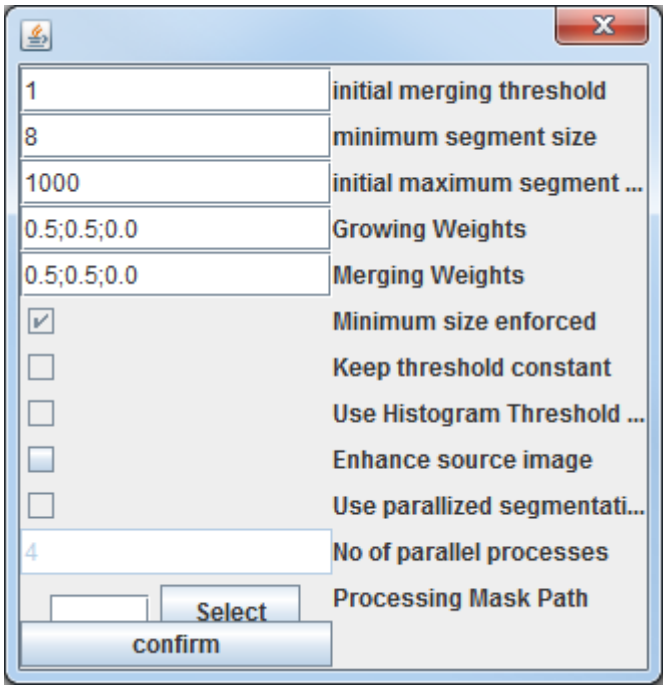


You should now see two more menus in the menu bar: **Segmentation** and **Edge Detection**. You should first go into edge detection and press the perform button. This will open a new dialog for the two parameters required for a canny based edge detector. Those are two threshold values. Set the **lower threshold** to 0.1 and the **upper threshold** to 0.2 and explore the result. These parameters are used for a gap-filling procedure during edge detection.



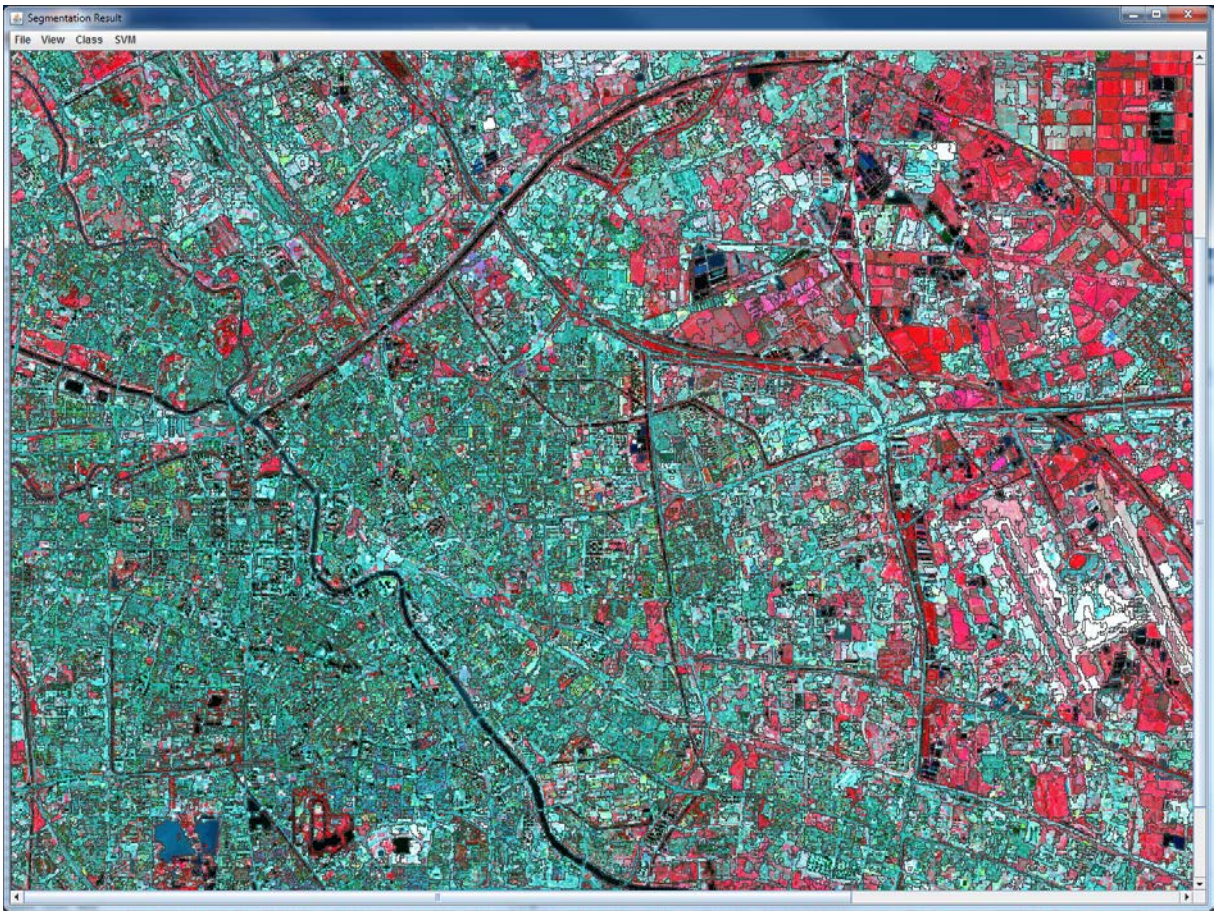
Usually the thresholds are selected in a way that the lower is about half of the upper threshold. You can run several edge detections consecutively, but only the last one performed will be used during segmentation. If you don't perform edge detection prior to segmentation the algorithm will run without the help of edges. Experiment a bit with the values to see the different effects of the edge detection algorithm. A good result of the edge detection process is when all major edges between different land cover classes are detected with a minimum of edges within a coherent area of the same class. When using the thresholds as above, too little edge is detected to build meaningful segments. Try to decrease the thresholds to obtain more edge.

After finishing the edge detection step a viewer with the detected edges will show up. Close the viewer, and then in the **Segmentation** menu click perform to start the segmentation process. Once you clicked on perform, the following dialog will show up:



In the dialog you can set the minimum and maximum segment size as parameters. When selecting your sizes keep in mind that usually over-segmented images (too many segments) give better classification results than under-segmented images (too few segments). When selecting your sizes, also think about the resolution of the data when deciding for your minimum segment size. (Tip: Set the minimum segment size to 8 and the initial maximum segment size to 1000 pixels). Once you have selected your sizes press **confirm** to let the segmentation run. This process will run about 10 minutes depending on your computer. *It is also recommended not to do anything else on the computer that is currently running the segmentation since it uses almost all of the computer's memory*. The smaller the minimum and maximum segment sizes become, the more time is required by the algorithm.

After the segmentation is finished a new window will appear with the segmentation results:



Explore the result by using the zoom button in the bar above (under view) and use the sliders to investigate the outcome. Try to experiment with minimum and maximum segments sizes **if** you examine the segments and feel that there are smaller features in the image that are currently mixed into segments that also contain other features, you might want to decrease minimum segment size. Or, on the other hand, if you feel that a lot of coherent features, e.g. agricultural areas that are segmented into too many small features, you might want to increase the maximum segment size. Try changing the values to differ 50 percent of the initially proposed ones (e.g. minimum segment size 4, 8, or 12 and maximum segment size 500, 1000 or 2000). The shapefile from the most reasonable segmentation should then be used for the training of the SVM classifier.

Training sample selection

Now we can start the classification process. Start to select your training segments for the Support Vector Machine (SVM)-classifier. Before you do so it is recommended to save the segments once. To do so open the **File** menu and select **export segment (shp)**. *It is very important that you include the .shp extension in you file name. It will not be automatically attached, and not writing it will result in an error*.

In the **class** menu you have to first select a class and then press the **select** button in the bottom. *Only when you hit the **select** button the class for labelling is actually selected*! For this training session, you should select at least 8 different land cover classes:

- roads
- bare/brown fields

- water
- tree cover
- airport runways
- vegetated (agriculturally used) fields
- see how many urban classes you can distinguish – think about roof colour and building structure. You should at least be able to distinguish two different urban classes: High density built-up areas with houses standing dense to each other and no green space and low density built-up building blocks with mostly residential function and greening in between.

Please use the first classes in the menu for this purpose because not consecutively selected class IDs can lead to problems in the SVM classifier (IMPORTANT: remember the order in which you select the classes (e.g. 3 = water).

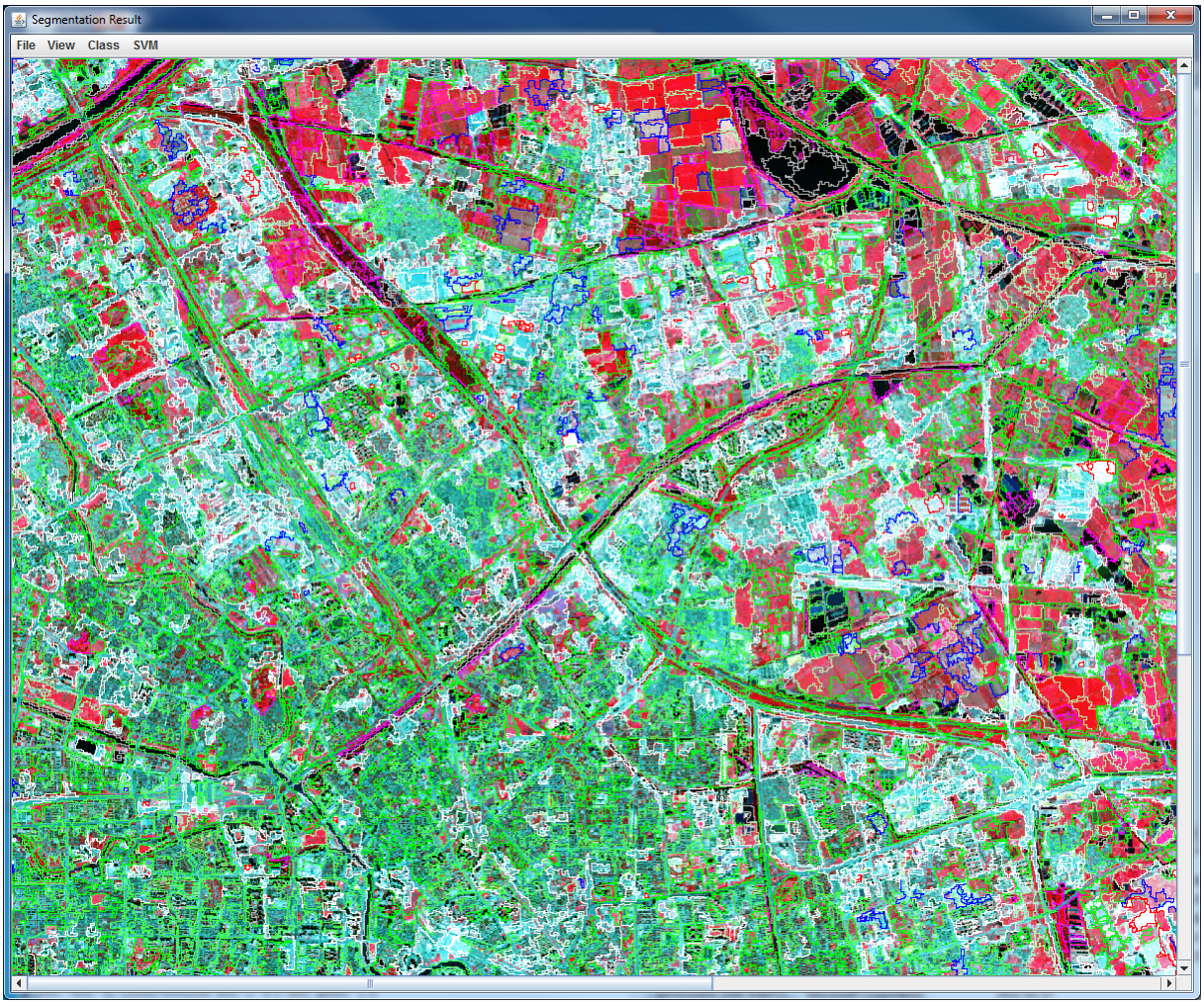
Labelling of a segment is done by right-clicking into a segment. If a segment was labelled wrong a click on the middle mouse button removes the label again. With a left click on a segment a dialog showing the statistics of a segment can be opened (observe that two statistics are recorded, i.e. mean and standard deviation of the pixels' digital numbers that compose the segment). Left clicking also highlights the outline of the segment.

Via the view menu it is possible to zoom in and out in the segmentation results. Scrolling over the map has to be done using the scrollbars in the bottom and right border of the GUI.

You should select at least 5 to 10 samples per class to make sure catch all instances of a class with an equal number of training samples per class. After having selected your segments it is recommended to once more export them to a shapefile. This enables you to recover your training segments in case you are not satisfied with your classification result and want to modify your training segments.

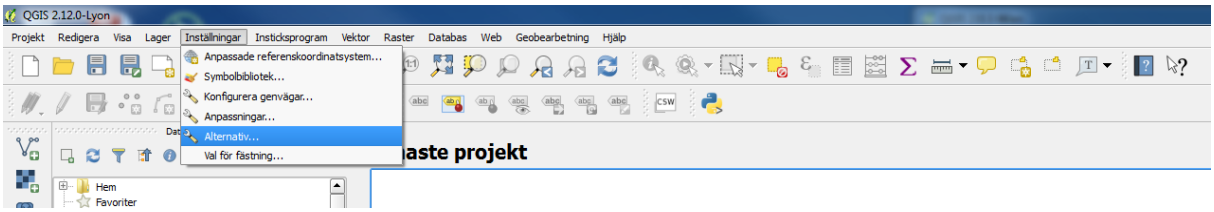
TIP: You can make use of high-resolution images over Tianjin provided in e.g. GoogleEarth or GoogleMaps if you are unsure about the classes and where there differences lie in order to make sure you label the classes correctly.

Classification
When you have all your samples selected **perform SVM** via the **SVM** menu. Just **confirm** the dialog and leave the parameters as they are. Now in the background a support vector machine is trained using your training samples. The SVM is based on a so called RBF-kernel and the two parameter fields you were seeing earlier are search ranges for finding the best possible combinations for the gamma and C parameter that you need to select for an RBF based SVM. The SVM takes as input parameters the mean and standard deviation of every segment for every band (those values that you can see when left clicking on a segment). Once the SVM is done (for this image it should only take a couple of seconds) save your segments once more, e.g. under *result.shp*. The result of the segmentation could now look similar to the following image:

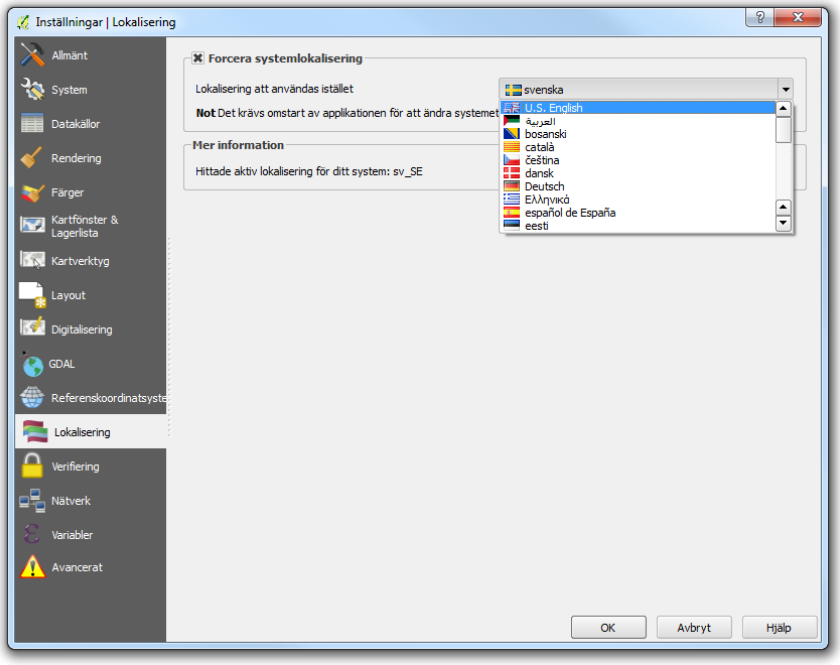

Segmentation Result
File View Class SVM

As it is currently not possible to change the predefined class colours in KTH-SEG, you should visualize the classified segments in another program. An appealing colour coding makes the result easier to interpret and eventually to be presented. For this purpose, start Quantum GIS Desktop (QGIS). QGIS is a powerful and easy-to-use open source GIS program that offers an alternative to commercial software packages. The latest version of QGIS can be obtained here: http://www.qgis.org/en/site/forusers/download.html

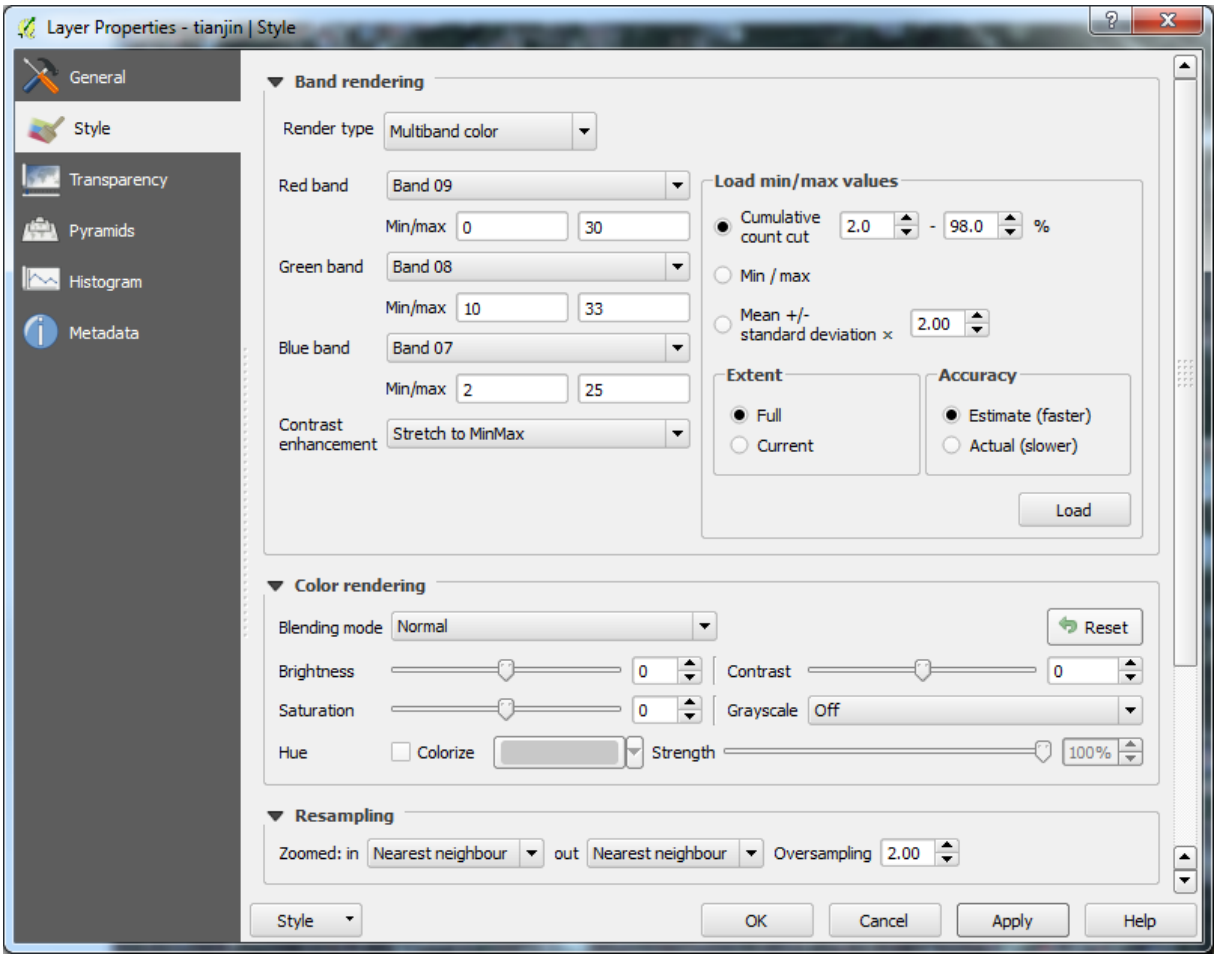
Start QGIS. If English is not the default language, change the language settings. You might also change the language to Chinese if you feel more comfortable with it. Go to Settings/Inställningar > Options…/Alternativ…



In the Locale/Lokalisering tab, change an appropriate language from the dropdown menu.

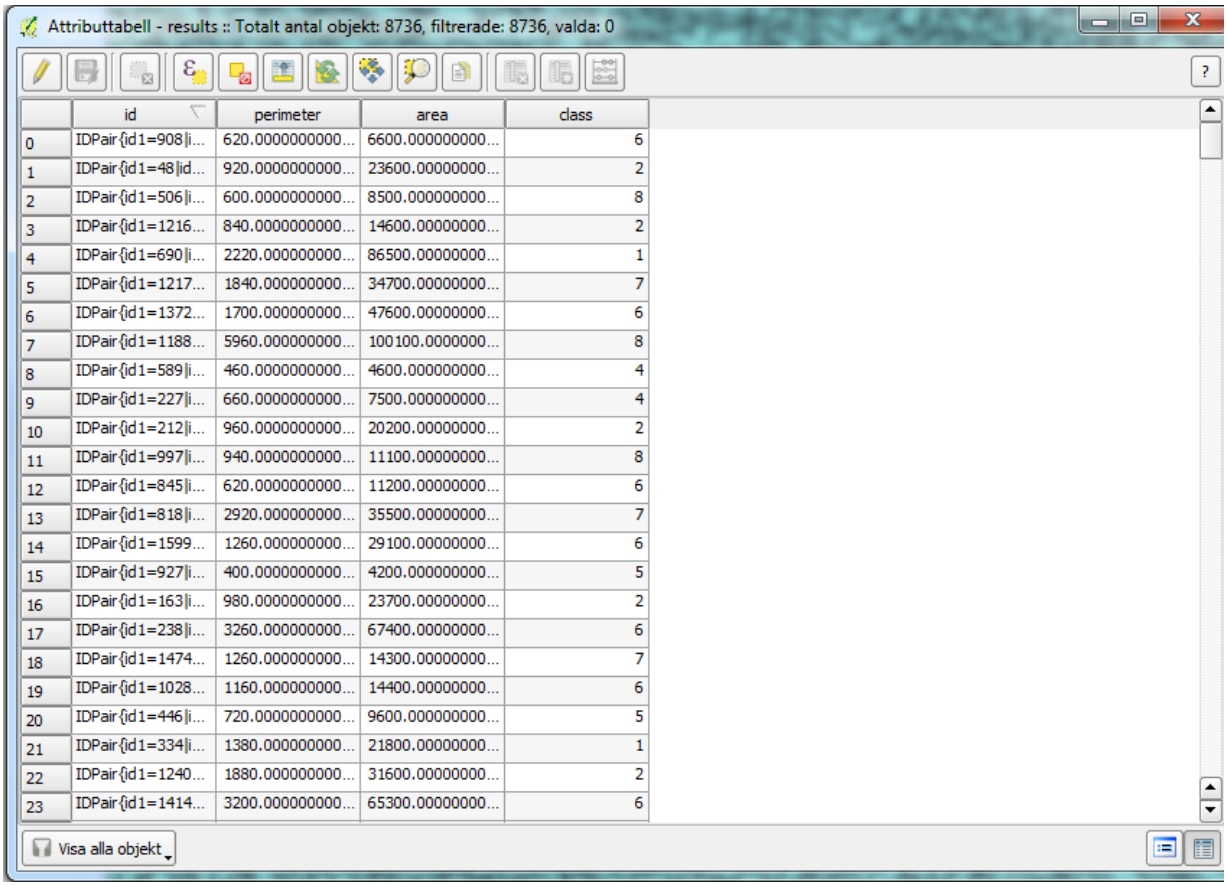


Begin with importing the *tianjin.tif* through choosing *Layer* > *Add Raster Layer*. Browse to your working folder, choose the image file and then *Open*. In order to create a true colour composite of the scene, right click on the layer in the left-hand menu and choose *Properties*. In the style tab, choose the band combination as in the following figure to display a RGB composite:



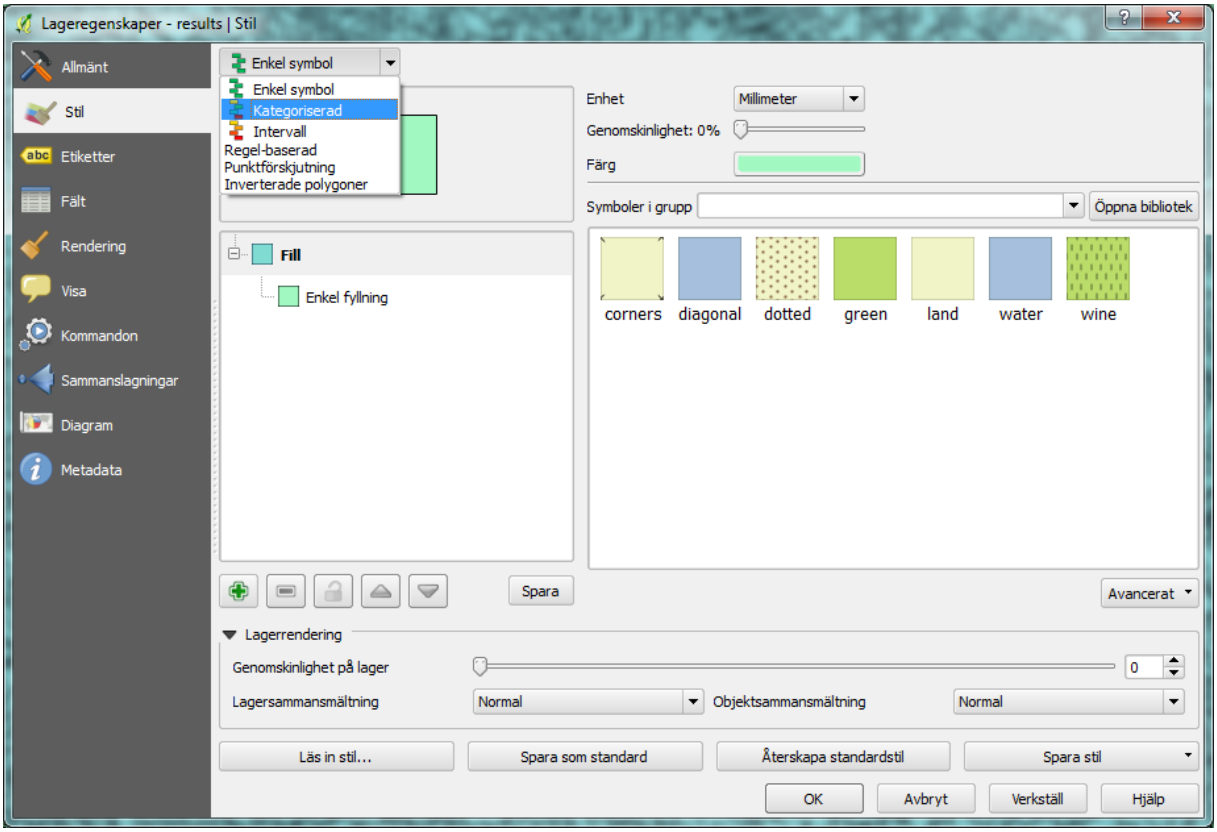
Now, overlay the vector layer. Click again on *Layer*, and add vector. Browse to your shapefile containing the classified segments, and then open.

Explore the attribute table of the newly added vector layer. Right click on the layer in the *Layer* menu to the left and choose *Open attribute table*. Observe that the file contains a column named class. The class number should correspond to the other you chose when selecting your training segments.

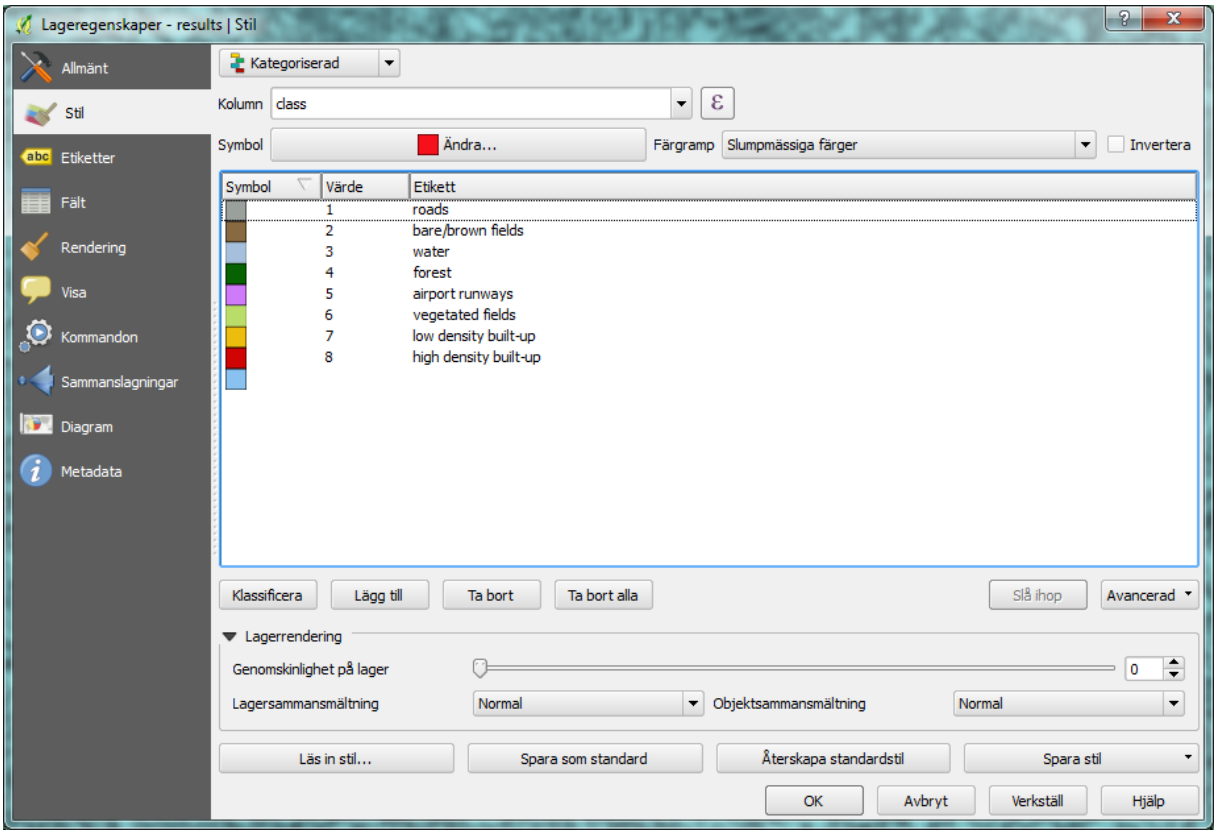


	id	perimeter	area	class
0	IDPair{id1=908|i...	620.0000000000...	6600.000000000...	6
1	IDPair{id1=48|id...	920.0000000000...	23600.00000000...	2
2	IDPair{id1=506|i...	600.0000000000...	8500.000000000...	8
3	IDPair{id1=1216...	840.0000000000...	14600.00000000...	2
4	IDPair{id1=690|i...	2220.000000000...	86500.00000000...	1
5	IDPair{id1=1217...	1840.000000000...	34700.00000000...	7
6	IDPair{id1=1372...	1700.000000000...	47600.00000000...	6
7	IDPair{id1=1188...	5960.000000000...	100100.0000000...	8
8	IDPair{id1=589|i...	460.0000000000...	4600.000000000...	4
9	IDPair{id1=227|i...	660.0000000000...	7500.000000000...	4
10	IDPair{id1=212|i...	960.0000000000...	20200.00000000...	2
11	IDPair{id1=997|i...	940.0000000000...	11100.00000000...	8
12	IDPair{id1=845|i...	620.0000000000...	11200.00000000...	6
13	IDPair{id1=818|i...	2920.000000000...	35500.00000000...	7
14	IDPair{id1=1599...	1260.000000000...	29100.00000000...	6
15	IDPair{id1=927|i...	400.0000000000...	4200.000000000...	5
16	IDPair{id1=163|i...	980.0000000000...	23700.00000000...	2
17	IDPair{id1=238|i...	3260.000000000...	67400.00000000...	6
18	IDPair{id1=1474...	1260.000000000...	14300.00000000...	7
19	IDPair{id1=1028...	1160.000000000...	14400.00000000...	6
20	IDPair{id1=446|i...	720.0000000000...	9600.000000000...	5
21	IDPair{id1=334|i...	1380.000000000...	21800.00000000...	1
22	IDPair{id1=1240...	1880.000000000...	31600.00000000...	2
23	IDPair{id1=1414...	3200.000000000...	65300.00000000...	6

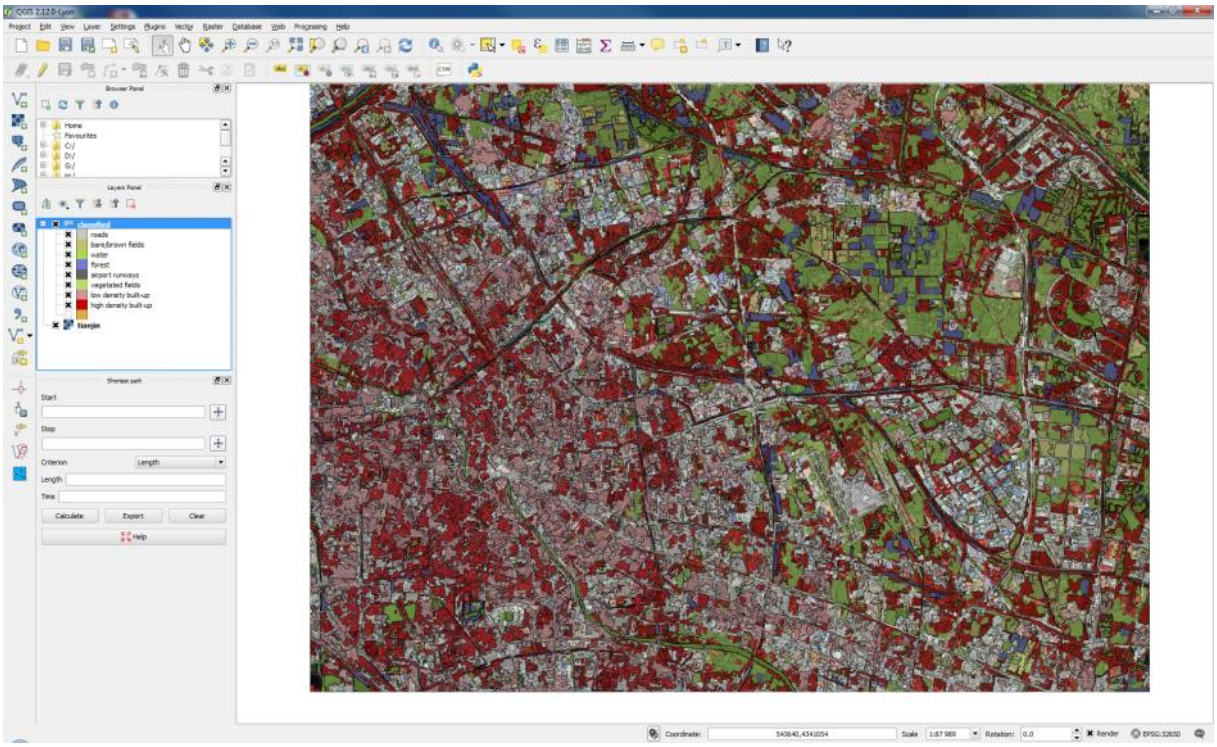
Close the attribute table and open the *Properties* of the shapefile layer instead. In the *Style* tab, choose *Categorized* instead of *Simple Symbol.*



Choose *class* from the *Column* menu and then *Classify*. Since the program does not know what each class number corresponds to, it assigns random colours to all classes. Choose a *Symbol* for each category (by double-clicking on the colour box) and give the class a correct name under the *Label* column.



Under *Layer Rendering*, you might want to make the layer somewhat transparent (e.g. 50%) so that you can see the underlying image at the same time. That gives you the possibility to investigate the classification result and the original data simultaneously and makes it easier to identify possible misclassifications. When you are satisfied with your choices, click on *Apply* and *Ok.* Explore the results.



Usually, you would also perform an accuracy assessment on the results to evaluate the correctness and reliability of the classification in other programs. To do this, you could simply convert the shapefiles into a raster surface based on the class descriptor field. Instead of performing an accuracy assessment at this point, you should instead think a bit about your results and the segmentation and classification process. In the following section you will find some questions that help you understand why the results turned out as they did and what could be done to further improve the results.

Afterthoughts and Questions

Inspect the classified image. Are you satisfied with the result? As no classification is perfect you will surely find some areas that are misclassified. Think about why this happened in respect to:

- how the classifier works (what features are used)
- your training sample selection
- how edge detection might influence the result
- how the segments are determined
- the parameter settings you chose
- the choice of classes

How would you try to improve the results? Think about:

- additional bands
- additional/alternative datasets to overcome limitations
- appropriate and maybe needed spectral and spatial resolutions
- features used in the classifier

References

Ban, Y., & Jacob, A. (2013). Object-based fusion of multitemporal multiangle ENVISAT ASAR and HJ-1B multispectral data for urban land-cover mapping. *Geoscience and Remote Sensing, IEEE Transactions on*, *51*(4), 1998-2006.