

Python, ERDAS & QGIS: (Geo-) Daten Handling

■ Übung 3

- Fähigkeiten von Python (z.B. „rasterio“ Bibliothek), ERDAS und QGIS kennenlernen
- Laden von (GeoTiff) Bildern (Python, QGIS & ERDAS)
- Benutzung Viewer (ERDAS & QGIS)
- Bänder Kombinieren (layerstack)
- Subsets erstellen mit Python (ev. ERDAS)
- Spektrale Profile erstellen (ERDAS oder QGIS)

Python Module für Fernerkundung und Bildverarbeitung

- Eine kurze Liste von wichtigsten Modulen:
 - **Pyproj** - Library for cartographic transformations.
 - **Cartopy** - A library providing cartographic tools.
 - **Basemap** - A matplotlib toolkit for plotting 2D data on maps.
 - **GDAL** - Geospatial Data Abstraction Library.
 - **Rasterio** - Expressing GDAL's data model using fewer non-idiomatic extension classes
 - **Shapely** - Creation, manipulation, and analysis of planar geometry objects.
 - **Pyshp** - Shapefile Library reads and writes ESRI Shapefiles.
 - **Fiona** - Fiona reads and writes spatial data files.
 - **LibLAS** - Reads and writes the LiDAR format.
 - **PDAL** - Library for translating and manipulating point cloud data.
 - **Pandas** - Python Data Analysis Library.
 - **GeoPandas** – Working with geospatial data + spatial operations.
 - **Seaborn** - Statistical data visualization.
 - **OpenCV** - A real time computer vision library.
 - **Scikit-image** - Image processing in Python.
 - **Scikit-learn** - Machine Learning in Python.
 - **Tensorflow** – Machine/Deep Learning in Python.
 - **PyTorch** - Machine/Deep Learning in Python.
 - **Keras** – Machine/Deep Learning Library.
 - ...

Übungsangaben

- Immer selbe Aufteilung
 - Lehrinhalte bzw. Lernziele
 - Daten
 - ... die in dieser Übung verwendet werden
 - Übungen
 - Auftrag
 - Auftrag in Worten
 - Darstellung der Ergebnisse (zur Betrachtung und für **Dokumentation**)
 - **Dokumentation**
 - Nur für Übungen die **abzugeben** sind (**N.N * Übung in ROT**)
 - Andere Übungen sind nur in wenigen Worten festzuhalten
 - Daten
 - die für diese Übung relevant sind.

Daten (1)

- Landsat TM 7 Bilder (2007)
 - **ORDNER: \data1\LS7\LS2007**
 - Namensformat: L7fppprrr_rrrYYYYMMDD_AAA.TIF
 - L71190027_02720020831_B10.TIF
 - L7 = Landsat 7
 - f = ETM+ data format = Level 1
 - ppp = starting PATH of the product
 - rrr_rrr = starting and ending ROW of the product
 - YYYYMMDD = acquisition date 2002 08 31
 - AAA = file type
 - B10 = Band 1
 - B61 = Band 6L (low gain)
 - B62 = Band 6H (high gain)
 - B80 = Band 8 (panchromatisch)
 - MTL = Level-1 metafile, GCP = Ground control points

Daten (2)

- Landsat TM 7 Bilder (2007) - \data1\LS7
 - geometrische Auflösung
 - B80 = 15 m
 - B10..B50 + B70 = 30 m
 - B61, B62 = 60 m
 - Bänder ETM+ MS/thermal
 - [1,2,3,4,5,6,7] = [BG,G,R,NIR,MIR(kurzw.),TIR,MIR(langw.)]
 - 8 = panchromatisch
 - Kanalkombinationen (BGR)
 - 123 = „True colour“
 - 234 = „False colour NIR“
 - 247 = „False colour MNIRG“
 - 345 = „False natural“ (ERDAS)
 - 346 = „Land Trafficability“ (ERDAS)

Übung 1 (1)

■ 3.1 Einlesen von GeoTIFF Bildern in Python

- Einlesen des panchromatischen Bildes (Landsat Bild, LS2007)
 - Metadaten abfragen:
 - Anzahl der Kanäle/ Bänder
 - Datentyp
 - Informationen zur Georeferenzierung (transform, crs,...)
 - Einlesen von Bilddaten („Pixeln“) eines spezifischen Bildkanals
 - Den Bildbereich [9800:11500,6500:8000] darstellen
 - Als Grauwertbild mit matplotlib Bibliothek
- Daten:
 - Input: \LS2007\ „panchromatischen Kanal“

Übung 1 (2)

■ 3.2 * Layerstacking (*rasterio, gdal, numpy*)

■ Auftrag

- Zusammenfügen der Kanäle [BG,G,R,NIR,MIR1,MIR2] in ein sechskanaliges Bild
- Darstellung (z.B. in ERDAS)
Kanalkombination RGB=[742]
(Achtung: **Kanalnummern von Landsat TM!**)
- **Dokumentation**
 - Python Code
 - Screenshot Kanalkombination (Übersicht und Detail Raum „Knoten Webling“, „Schwarzlteiche“)

■ Daten

- Input: \LS2007\... _bNN.tiff
- Output (Vorschlag): ls2007_ms_notir.img

Übung 1 (3)

■ 3.3 Subsets (*rasterio, gdal, numpy*)

■ Auftrag

- Bilden von 18 km x 18 km Subsets aus den LS 2007 Bildern bei gegebener Linken Oberen Ecke (upper left corner, ULC)
- Darstellen (*ERDAS*): Pan als Greyscale, MS als Falschfarb-Infrarotbild
- Dokumentation
 - QGIS Screenshots + Python Code

■ Daten

- Input:
 - ls2007_pan
 - ls2007_ms_notir.img
- Output (Suffix):
 - ..._subpcNN.img

Übung 1 (4)

■ 3.3 Subsets

■ ULC für Subsets

- *Gruppe NN: Easting(X), Northing(Y)*
- *Gruppe 1: 511920,5344920*
- *Gruppe 2: 511920,5326920*
- *Gruppe 3: 511920,5308920*
- *Gruppe 4: 511920,5290920*
- *Gruppe 5: 511920,5272920*
- *Gruppe 6: 511920,5254920*
- *Gruppe 7: 511920,5236920*
- *Gruppe 8: 511920,5218920*
- *Gruppe 9: 511920,5200920*

Übung 1 (5)

- 3.4 * Spektrale Profile (*ERDAS oder QGIS-Semi-Automatic Classification Plugin*)
 - Auftrag
 - Erstellung von Spektralen Profilen für fünf (5) Landbedeckungsklassen (engl. landcover, kurz LC)
 - z.B. Wald, Ackerland, Siedlung, Wasser, Grünland, Schnee ...
 - Punkte kleinräumig wählen, so dass auf dem Bild noch etwas erkennbar ist!
 - Dokumentation
 - Screenshot mit Lage der Punkte (**sollen erkennbar sein!**). Dazu als Hilfe shp-File verwenden.
 - Spectral Profile Chart
 - Beurteilung in welchen Bändern die LC-Klassen trennbar sind (mit Begründung)
 - Daten
 - Input:
 - ls2007_ms_notir_subNN.img