**Runnable on Spyder/Jupyter Notebook (Python Platforms)**

**Instruction/Coding - 1**

**To search, download, read, write and do basic processing on satellite images (Landsat)**

**# You do not need to import all the libraries, however, you need to install most of them to be able to import**

**import os**

**import pandas as pd**

**import geopandas as gpd # the GEOS-based vector package**

**from geopandas.geodataframe import GeoDataFrame**

**import numpy as np # the array computation library**

**import matplotlib as mp**

**import matplotlib.pyplot as plt # the visualization package**

**import gmplot**

**import rasterio as rio # the GEOS-based raster package**

**from rasterio.plot import show**

**from rasterio.plot import show\_hist**

**from shapely.geometry import Polygon**

**from geopandas import GeoSeries**

**from shapely.geometry import Point**

**from shapely.affinity import translate**

**from rasterio import plot as rioplot**

**# import elevation**

**import seaborn as sns # for scientific graphs**

**from rasterio.transform import from\_bounds, from\_origin**

**from rasterio.warp import reproject, Resampling**

**from scipy.stats import linregress**

**from scipy.io import arff**

**from numpy import asarray**

**from numpy import save**

**from numpy import load**

**from io import StringIO**

**import earthpy as et**

**import earthpy.spatial as es**

**import earthpy.plot as ep**

**import gmaps**

**Links to search and download Landsat Images**

* [EarthExplorer (usgs.gov)](https://earthexplorer.usgs.gov/)
* [**https://livingatlas2.arcgis.com/landsatexplorer/**](https://livingatlas2.arcgis.com/landsatexplorer/)
* [Link c58ecc2eb008b710aa6a16dc827beaca - Earth Engine Code Editor (google.com)](https://code.earthengine.google.com/c58ecc2eb008b710aa6a16dc827beaca)

**#For Georgian college-Imgage processing**

**# If you have anu questions, pleas contact me, alireza.ghaffari@georgian.college.ca**

**# Change the path and directories to you own path**

**# Read and plot raster data-Landsat Data**

l7\_composite = r"C:\geopandas\Toronto\_Landsat\l7\_composite.img"

raster = rio.open(l7\_composite)

band1 = raster.read(1)

show(band1)

**# Show band by band**

show((raster, 4), cmap='Reds')

show((raster, 3), cmap='Greens')

show((raster, 1), cmap='Blues')

show((raster, 1))

**# Calculate bands stats**

array = raster.read()

stats = []

for band in array:

stats.append({

'min': band.min(),

'mean': band.mean(),

'median': np.median(band),

'max': band.max()})

print(stats)

**# Create histogram**

show\_hist(raster, bins=50, lw=0.0, stacked=False, alpha=0.3,

histtype='stepfilled', title="Histogram")

**# Read raster data**

# l7\_composite = r"C:\Ali\geopandas\Toronto\_Landsat\l7\_composite.img"

raster = rio.open(l7\_composite)

#Read the grid values into numpy arrays

nir = raster.read(4)

red = raster.read(3)

green = raster.read(2)

# Create the false composite by stacking

nrg = np.dstack((nir, red, green))

# Let's see how our false color composite looks like

plt.imshow(nrg

**# Convert to floats**

red = red.astype('f4')

nir = nir.astype('f4')

np.seterr(divide='ignore', invalid='ignore')

**# Calculate NDVI using numpy arrays**

ndvi = (nir - red) / (nir + red)

#Plot the NDVI

plt.imshow(ndvi, cmap='terrain\_r')

# Add colorbar to show the index

plt.colorbar()

**# Create the true composite by stacking**

red1 = raster.read(3)

green1 = raster.read(2)

blue1 = raster.read(1)

# Create the true composite by stacking

rgb = np.dstack((red1, green1, blue1))

# Let's see how our true color composite looks like

plt.imshow(rgb)

**Continued to Instruction/Coding - 2**

**To apply Artificial Neural Network to Classify Landsat Image**