Special Topics in GIS Proposal

Owen Smith, 2020
Dr. Huidae Cho

Title:

A Reproducible Supervised Classification System for Tree Canopy and Deforestation Detection Within an Open Source Python Framework Utilizing NAIP Imagery. A Case Study of Georgia

1. Objectives:

1. Utilize supervised classification algorithms from Scikit-learn and automated ARVI calculations of 4-band 1m NAIP imagery to create a reproducible method to classify canopy/forest to aid in monitoring deforestation.
2. Create a Python library designed to optimize classification of raster data in a geospatial setting
3. If time permits, develop supervised classification algorithm outside of Scikit-learn for method.
4. Create necessary data to use in mitigating deforestation.
5. Case study Georgia
6. Publish paper detailing work

2. Overview:

2.1 Background:

Deforestation monitoring is an essential part of maintaining any environment as the loss of forested lands leads to increased CO2 being placed into the atmosphere while simultaneously eliminating carbon storage (Bala, Govindasamy, et al. 2007). At smaller scales it leads to both increased runoff rates and subsequently increased erosion, especially in areas where no plant reclamation is initiated (Benito, E., et al, 2003). Accurately monitoring deforestation to mitigate these effects on a large scale can be a time consuming and difficult process to complete (Basu, Saikat, et al. 2015). Furthermore, commercial software dedicated to completing these tasks such as eCognition (Trimble Inc.) or Textron Systems Feature Analyst (Textron Systems 2010) can be expensive with little insight into how their algorithms are truly working as they are closed source. The lack of knowledge into the inner workings of commercial software leads to the consideration of applying any number of existing open source libraries (Sonnenburg 2007). Contrasting with closed source software, open source work allows for the collaboration and modification of projects between others to serve different needs and purposes, and subsequently allows for transparency in research that leads to increased reproducibility and access (Sonnenburg 2007).

Previous studies have utilized open source libraries such as Keras-TensorFlow, PyTorch, and the Orfeo Toolbox API [Application programming interface] for landcover classification and deforestation monitoring (Abujayyab & Karaş 2019, Anh et al. 2019, Rakshit et al. 2018, Grings et al. 2019). However, not much research has extensively utilized Scikit-learn. One study has used Scikit-learn for deforestation monitoring in which a committee system was developed using Scikit-learns k-Nearest Neighbors, linear Discriminant and Multi-layer perceptron (Dallaqua et al. 2018). The committee method however is not paired with a reproducible method that can be made scalable. Other research utilizing Scikit-learn has used Scikit-learn algorithms, logistic regression and support vector machines [SVM], as baselines or for comparing with other developed machine learning algorithms, but again no method for scalable reproducibility was applied (Šimić de Torres 2016, He et al. 2017, Ortega et al. 2019).

Scikit-learn is built on top of NumPy (van der Walt et al. 2011) and SciPy (Vertanen et al. 2019), two extensive Python libraries which are easy to utilize (Pedregosa et al. 2011) and will subsequently allow for the optimization of raster analysis using GDAL, a Python library used for geospatial data processing (Warmerdam 2008) While other open source packages like PyTorch and TensorFlow have faster computing times due to their ability to run parallel on graphics processing units [GPU] opposed to the central processing unit [CPU] , Scikit-learn is only capable of parallel computation on CPU's. However, PyTorch and TensorFlow are currently only capable of GPU parallel computation on Nvidia brand GPU's as they are built around Nvidia's proprietary parallel computing platform (GPU Support | TensorFlow 2020, CUDA Semantics – PyTorch 2019), called Compute Unified Device Architecture [CUDA] (Nickolls 2008). The use of CUDA in other Machine Learning packages effectively eliminates other graphics cards such as those produced by Intel or AMD and would be counterproductive to the aim of producing a reproduceable open source classification system capable of utilization regardless of hardware (Nickolls 2008). In another effort to increase computing times Scikit-learn is also built utilizing Cython, allowing it to reach performance levels of compiled languages (Pedregosa et al. 2011, Behnel 2011). Additionally, the thorough documentation Scikit-learn possesses along with its extensive supervised algorithms makes it an ideal choice for creating a process that can be repeatable and reproducible (Pedregosa et al. 2011).

Studies in the past which utilize open source classification libraries have also not been able to achieve accuracy at a level which 1-meter NAIP [National Agricultural Imagery Program] imagery can provide, having been limited to using only public access imagery such as Landsat-8 which consists of 30-meter resolution bands or MODIS which contains resolutions ranging from 250 meters to 1,000 meters (Šimić de Torres 2016, Dallaqua et al. 2018). Even though NAIP imagery is on a 3-year cycle and cannot match the temporal frequency in which satellite imagery is taken, NAIP imagery is taken during seasons in which agriculture is growing in the United States ensuring similar characteristics between datasets (NAIP 2009, NAIP Imagery 2019). Furthermore, clouds masks will not be needed for processing as NAIP imagery's quality control removes any image that has more than 10% cloud cover per quarter quad rendering the need for a cloud mask negligible (NAIP 2009, NAIP Imagery 2019).

The availability of 4 band NAIP datasets containing red, green, blue, and near infrared bands allows for the creation of an efficient automated process to calculate vegetation indices which will then be fed into the supervised classification algorithm (USDA, 2009). The lack of clouds mentioned previously in NAIP imagery will also remove issues previous studies had with utilizing vegetation indexes as the presence of clouds would render the index increasingly unreliable the more cloud cover the scene contained (Li et al. 2004). The vegetation index chosen to use is the atmospherically resistant vegetation index [ARVI] (Kaufman & Tanre 1992)(Figure 1.) which was chosen over other vegetation indices such as the Normalized Difference Vegetation Index [NDVI] (Tucker 1979) or the Enhanced Vegetation Index [EVI] (Jiang et al. 2007) as the ARVI corrects atmospheric scattering and has been shown to perform better than other vegetation indices. (Liu et al. 2004)

Figure 1.

$$ARVI = \frac{NIR - (2 * Red) + Blue}{NIR + (2 * Red) + Blue}$$

To supplement the ARVI, the use of the visible atmospheric resistant index [VARI] (Figure 2.) will be explored (Gitelson et al. 2002). The VARI only uses the bands of the visible spectrum which enables it to be used with imagery containing only three bands, and like the ARVI looks to mitigate atmospheric effects (Gitelson et al. 2002). If the performance of the VARI is at an acceptable level, then it will allow for areas which may not have access to high quality four band imagery to be properly monitored, and for the use of 3 band NAIP imagery which is easier to acquire.

Figure 2.

$$VARI = \frac{Green - Red}{Green + Red - Blue}$$

Previous research has not effectively published a reproduceable and inexpensive method for deforestation monitoring as researchers often do not make available the framework with which the research was conducted (Sonnenburg 2007). The goal of the proposed research, however, is to create a process that can be repeated to effectively monitor deforestation. With the benefits NAIP imagery possess combined with the effectiveness of Scikit-learn a dedicated remote sensing method can be developed to solve these issues.

2.2 Process/Method:

What is proposed is a fully open source supervised classification system designed specifically to create an efficient canopy/forest classification system using NAIP imagery. The process will involve automating ARVI or VARI calculations directly from the imagery without any preprocessing. Automation will be completed by reading each band as a NumPy array into Python using GDAL (GDAL/OGR Contributors 2019). These NumPy arrays will then be fed into the calculation to produce the ARVI/VARI raster, which will then be subsequently saved as a float32 tiff. The output tiff files will be what is classified by the supervised classification algorithm. Early calculations of individual ARVI/VARI raster's on Landsat-8 raster images are taking an average of 4 seconds to compute, so the computation time of each NAIP quarter quad will be significantly shorter as Landsat-8 scenes are considerably larger than the 3.75' x 3.75' NAIP tiles (NAIP Imagery 2020).

Training data will then be created in an open source geospatial software, Quantum GIS [QGIS] (QGIS Development Team 2020) and consist of two classes. Class zero being non tree, and class one being trees. The training data will then read by GDAL, rasterized and then converted into a NumPy array which will be used to develop training labels in the Scikit-learn algorithm.

Scikit-learn models will be trained and a baseline for number of samples needed will be determined in order to aid in future use. After training models and gathering baseline data of performance and accuracy, the best performing model will be chosen. A key factor considered when choosing models will be their ability to run parallel on the CPU.

Using the chosen Scikit-learn model a Python library will be created to combine the raster classification and supervised learning into one process. The combined process will firstly

read the bands of each NAIP quarter quad tile as NumPy arrays and perform the required index calculations. The resulting index calculations will then be fed into the chosen supervised classification model along with the training data. The model will return an accuracy score determined either by one of Scikit-learns preexisting score algorithms or by creating a custom score function. The model score will allow the user to decide if data needs to be retrained or not.

Scalability regarding how large of batch sizes are ideal will need to be determined as well. Currently, separating the imagery by physiographic region seems to work adequately, but it also leads to large differences in processing times between regions. A standardized method not reliant upon external geospatial data to equally distribute imagery into batches would be preferable but need to be developed.

3. Tentative Timeline:

Month	**Start Date**	**End Date**	**Task**
January	1/12/2020	1/19/2020	Planning
	1/12/2020	1/26/2020	Literature review
	1/26/2020	2/2/2020	Automate ARVI & VARI calculations
February	2/2/2020	3/2/2020	Train Scikit models & Develop scoring system
March	3/2/2020	3/22/2020	Combine ARVI automation with Scikit model
	3/22/2020	3/29/2020	Compare results with unsupervised class and/or GFC project
	3/29/2020	4/19/2020	Complete report / Review

* Weekly meetings every Monday from 11:00am to 12:00pm

References:

Abujayyab, S. K., & Karaş, I. R. (2019). GEOSPATIAL MACHINE LEARNING DATASETS STRUCTURING AND CLASSIFICATION TOOL: CASE STUDY FOR MAPPING LULC FROM RASAT SATELLITE IMAGES. International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences.

Anh, N. D., Tuan, V. A., & Hang, N. T. Deforestation hot-spot extraction from Radar Change Ratio (RCR) analysis of Sentinel-1 time series data in Dak G'Long district, Dak Nong province.

Bala, G., Caldeira, K., Wickett, M., Phillips, T. J., Lobell, D. B., Delire, C., & Mirin, A. (2007). Combined climate and carbon-cycle effects of large-scale deforestation. Proceedings of the National Academy of Sciences, 104(16), 6550-6555.

Basu, S., Ganguly, S., Nemani, R. R., Mukhopadhyay, S., Zhang, G., Milesi, C., … & Cook, B. (2015). A semiautomated probabilistic framework for tree-cover delineation from 1-m NAIP imagery using a high-performance computing architecture. IEEE Transactions on Geoscience and Remote Sensing, 53(10), 5690-5708.

Benito, E., Santiago, J. L., De Blas, E., & Varela, M. E. (2003). Deforestation of water-repellent soils in Galicia (NW Spain): effects on surface runoff and erosion under simulated rainfall. Earth Surface Processes and Landforms: The Journal of the British Geomorphological Research Group, 28(2), 145-155.

CUDA semantics — PyTorch master documentation. (2019). Retrieved January 22, 2020, from https://pytorch.org/docs/stable/notes/cuda.html

Dallaqua, F. B., Faria, F. A., & Fazenda, A. L. (2018, October). Active Learning Approaches for Deforested Area Classification. In 2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI) (pp. 48-55). IEEE.

GDAL/OGR contributors (2019). GDAL/OGR Geospatial Data Abstraction software Library. Open Source Geospatial Foundation. URL https://gdal.org

Gitelson, A. A., Stark, R., Grits, U., Rundquist, D., Kaufman, Y., & Derry, D. (2002). Vegetation and soil lines in visible spectral space: a concept and technique for remote estimation of vegetation fraction. International Journal of Remote Sensing, 23(13), 2537-2562.

GPU support | TensorFlow. (2020, January 13). Retrieved January 22, 2020, from https://www.tensorflow.org/install/gpu

Grings, F., Roitberg, E., & Barraza, V. (2019). EVI Time-Series Breakpoint Detection Using Convolutional Networks for Online Deforestation Monitoring in Chaco Forest. IEEE Transactions on Geoscience and Remote Sensing.

Jiang, Z., Huete, A. R., Didan, K., & Miura, T. (2008). Development of a two-band enhanced vegetation index without a blue band. Remote sensing of Environment, 112(10), 3833-3845.

Kaufman, Y. J., & Tanre, D. (1992). Atmospherically resistant vegetation index (ARVI) for EOS-MODIS. IEEE transactions on Geoscience and Remote Sensing, 30(2), 261-270.

Li, M., Liew, S. C., & Kwoh, L. K. (2004, July). Automated production of cloud-free and cloud shadow-free image mosaics from cloudy satellite imagery. In Proceedings of the XXth ISPRS Congress (pp. 12-13).

Liu, G. R., Liang, C. K., Kuo, T. H., & Lin, T. H. (2004). Comparison of the NDVI, ARVI and AFRI vegetation index, along with their relations with the AOD using SPOT 4 vegetation data. Terrestrial, Atmospheric and Oceanic Sciences, 15(1), 15-31.

NAIP Imagery. (n.d.). Retrieved January 21, 2020, from https://www.fsa.usda.gov/programs-and-services/aerial-photography/imagery-programs/naip-imagery/

NAIP. Retrieved January 21, 2020, from http://www.fsa.usda.gov/Internet/FSA_File/naip_2009_info_final.pdf

Nickolls, J., Buck, I., Garland, M., & Skadron, K. (2008). Scalable parallel programming with CUDA. Queue, 6(2), 40-53.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E.A. Quintero, Charles R Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. (2019) SciPy 1.0–Fundamental Algorithms for Scientific Computing in Python.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of machine learning research, 12(Oct), 2825-2830.

QGIS Development Team (2020). QGIS Geographic Information System. Open Source Geospatial Foundation Project. http://qgis.osgeo.org

Rakshit, S., Debnath, S., & Mondal, D. (2018). Identifying Land Patterns from Satellite Imagery in Amazon Rainforest using Deep Learning. arXiv preprint arXiv:1809.00340.

Sonnenburg, S., Braun, M. L., Ong, C. S., Bengio, S., Bottou, L., Holmes, G., ... & RÃĪtsch, G. (2007). The need for open source software in machine learning. Journal of Machine Learning Research, 8(Oct), 2443-2466.

Stefan Behnel, Robert Bradshaw, Craig Citro, Lisandro Dalcin, Dag Sverre Seljebotn and Kurt Smith. Cython: The Best of Both Worlds, Computing in Science and Engineering, 13, 31-39 (2011), DOI:10.1109/MCSE.2010.118

Stéfan van der Walt, S. Chris Colbert and Gaël Varoquaux. The NumPy Array: A Structure for Efficient Numerical Computation, Computing in Science & Engineering, 13, 22-30 (2011), DOI:10.1109/MCSE.2011.37

Textron Systems (2010) Feature Analysist Release 5.0 Providence, RI

Trimble Inc. (2019) eCognition Release 1.3 Sunnydale, CA

Tucker, C. J. (1979). Red and photographic infrared linear combinations for monitoring vegetation. Remote sensing of Environment, 8(2), 127-150.

Warmerdam, F. (2008). The geospatial data abstraction library. In Open source approaches in spatial data handling (pp. 87-104). Springer, Berlin, Heidelberg.

Šimić de Torres, I. (2016). Analysis of satellite images to track deforestation (Bachelor's thesis, Universitat Politècnica de Catalunya).