The Journal of Open Source Software

CoreBreakout: Subsurface Core Images to Depth-Registered Datasets

Ross G. Meyer[1], **Thomas P. Martin**[1], **and Zane R. Jobe**[1]

1 Department of Geology and Geological Engineering, Colorado School of Mines

DOI: 10.21105/joss.01969

Software

- Review
- Repository
- Archive

Editor: Katy Barnhart

Reviewers:

- @brendonhall
- @JesperDramsch
- @jessepisel

Submitted: 09 December 2019
Published: 19 June 2020

Summary

Core samples – cylindrical rock samples taken from subsurface boreholes – are commonly used by Earth scientists to study geologic history and processes. Core is usually cut into one-meter segments, slabbed lengthwise to expose a flat surface, and stored in cardboard or wooden boxes which are then photographed to enable remote inspection. Unlike other common sources of borehole data (e.g., well logs Rider & Kennedy, 2011), core is the only data that preserves true geologic scale and heterogeneity.

A geologist will often describe core by visual inspection and hand-draw a graphic log of the vertical changes in grain size and other rock properties (e.g., Jobe et al., 2017). This description process is time consuming and subjective, and the resulting data is analog. The digitization and structuring of core image data allows for the development of automated and semi-automated workflows, which can in turn facilitate quantitative analysis of the millions of meters of core stored in public and private repositories around the world.

`corebreakout` is a Python package that provides two main functionalities: (1) a deep learning workflow for transforming raw images of geological core sample boxes into depth-registered datasets, and (2) a `CoreColumn` data structure for storing and manipulating the depth-registered image data. The former uses the Mask R-CNN algorithm (He, Gkioxari, Dollár, & Girshick, 2017) for instance segmentation, and is built around the open source TensorFlow and Keras implementation released by Matterport, Inc. (Abdulla, 2017).

Mask R-CNN Workflow

The primary user workflow enabled by `corebreakout` is depicted in Figure 1. It is straightforward for geologists to add their own labeled training images using `LabelMe` (Russell, Torralba, Murphy, & Freeman, 2007; Wada, 2016), configure and train new Mask R-CNN models on the labeled images, and subsequently use the trained models to process their own unlabeled images and compile depth-aligned datasets.

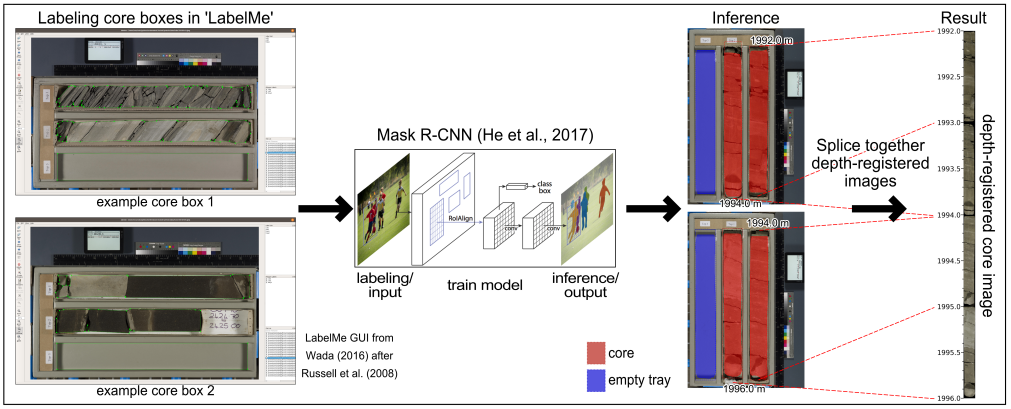

Figure 1: Primary User Workflow

In the future, we would like to train a more generalized model, but for now we anticipate that most users will have to train their own segmentation models. We have found labeling 25-30 images to be the point of diminishing returns for segmentation accuracy, but this number is likely dependent on the consistency of image layout and core material within a given dataset.

Trained models can be loaded using the `CoreSegmenter` class, which provides methods for processing images using the model and according to user-specified layout parameters.

`CoreColumn` Data Structure

The other main piece of functionality provided by `corebreakout` is the `CoreColumn` class, which is a container for depth-registered, single-column images of core material, allowing for joint manipulation of images and associated depth arrays. `CoreColumns` may be sliced, stacked, and iterated over, and they include saving, loading, and plotting functionality. Usage details can be found in the documentation and the provided `CoreColumn` tutorial.

General Functionality

`corebreakout` supports standard vertical and horizontal core image layouts, and provides several methods for measuring and assigning depths to core sample columns, including by labeling arbitrary "measuring stick" objects (e.g., rulers, empty trays). We provide a labeled dataset courtesy of the British Geological Survey's OpenGeoscience project, as well as a Mask R-CNN model trained on this dataset for testing and demonstration.

In addition to the core Python package, the source code includes scripts for training models, extracting text meta-data from images with optical character recognition (Smith, 2007), and processing directories of images with saved models.

`corebreakout` has been used to compile a large image dataset for ongoing work in image-based lithology classification (Martin, Meyer, & Jobe, 2019). We plan to release our modeling code as a separate project that uses the `CoreColumn` data structure to combine depth-registered image data, sampled well log data, and interval labels into multi-modal datasets for sequence prediction.

Acknowledgements

We would like to acknowledge the contribution of open source subsurface core images from the British Geological Survey (https://bgs.ac.uk/), and financial support from Chevron through the Chevron Center of Research Excellence at the Colorado School of Mines (https://core.mines.edu/).

References

Abdulla, W. (2017). Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow. *GitHub repository*. https://github.com/matterport/Mask_RCNN.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. B. (2017). Mask R-CNN. *CoRR*, *abs/1703.06870*. Retrieved from http://arxiv.org/abs/1703.06870

Jobe, Z., Sylvester, Z., Pittaluga, M. B., Frascati, A., Pirmez, C., Minisini, D., Howes, N., et al. (2017). Facies architecture of submarine channel deposits on the western Niger Delta slope: Implications for grain-size and density stratification in turbidity currents. *Journal of Geophysical Research: Earth Surface*, *122*(2), 473–491. doi:10.1002/2016JF003903

Martin, T., Meyer, R., & Jobe, Z. (2019). Lithology prediction of slabbed core photos using machine learning models. doi:10.6084/m9.figshare.8023835.v2

Rider, M., & Kennedy, M. (2011). *The Geological Interpretation of Well Logs: Third Edition*. Rider-French Consulting Limited. ISBN: 978-09541906-8-2

Russell, B. C., Torralba, A., Murphy, K. P., & Freeman, W. T. (2007). LabelMe: A database and web-based tool for image annotation. *International Journal of Computer Vision*, *77*(1-3), 157–173. doi:10.1007/s11263-007-0090-8

Smith, R. (2007). An overview of the Tesseract OCR engine. In *ICDAR '07: Proceedings of the Ninth International Conference on Document Analysis and Recognition* (pp. 629–633). Washington, DC, USA: IEEE Computer Society. ISBN: 0-7695-2822-8

Wada, K. (2016). labelme: Image Polygonal Annotation with Python. *GitHub repository*. https://github.com/wkentaro/labelme.