



MACHINE LEARNING FOR GEOSCIENCE

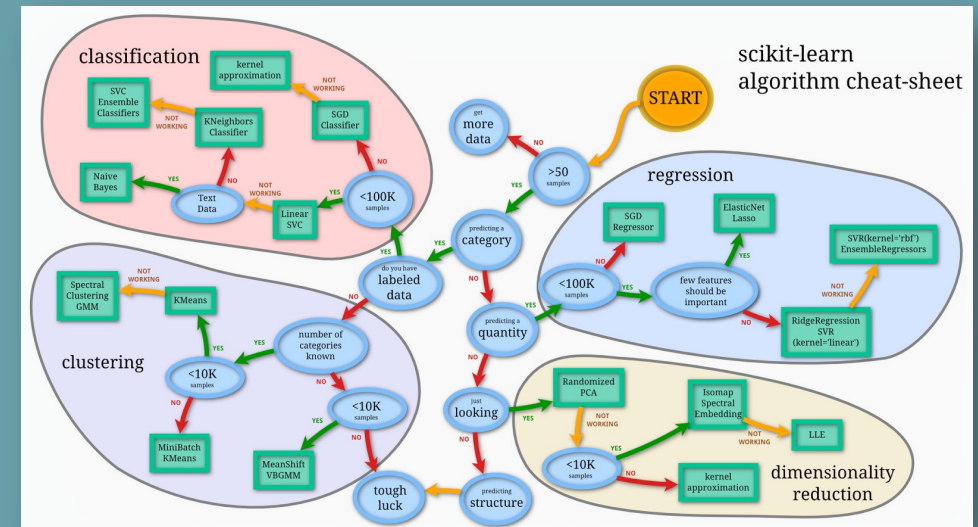
Nathalie Redick, Matthew Tarling, James Kirkpatrick

BACKGROUND



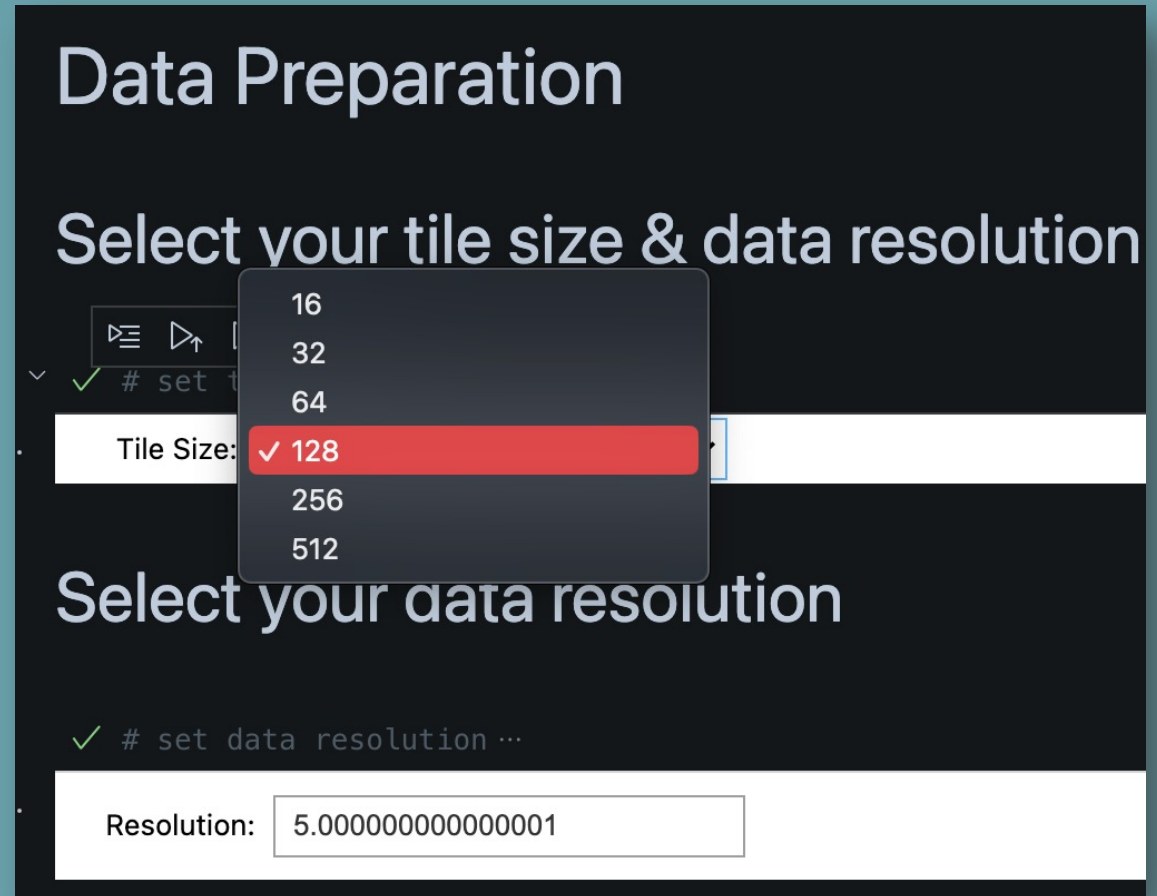
Why Machine Learning?

- What is machine learning (ML)?
 - *Very simply, it's type of statistical analysis that "trains" on data in a way that imitates the way that humans learn*
- Machine learning can support research by speeding up complex calculations, providing new insights, automating tedious tasks, etc.
- However, knowing how and when to use machine learning can be difficult for researchers outside of computer science
- Most large ML libraries (Tensorflow, Keras, PyTorch) either do not have any functionality for dealing with geospatial data, or very little (with poor documentation)
 - *This means writing a lot of custom functions and extra data cleaning*



Objective

- Design an end-to-end workflow that handles everything from data preprocessing to training a model, while allowing the user to choose parameters using buttons & dropdown menus, etc.
- A step-by-step guide explaining how & why to adjust the parameters will be provided alongside the code



Sample Applications

- Automatically identifying hurricanes in satellite imagery
- Determining crop types
- Mapping roads
- Finding fault scarps
- ...

HOW DOES IT WORK?



The Workflow

- The code is written in Python in a Jupyter Notebook (code cells that can be run individually, similar to using %% for sections in MATLAB)
 - *Outputs of each cell are visible beneath the cell*

▼ Add Data

Upload your data

Features

Features are the inputs the model learns in order to predict a *mask*. For example, if you want to predict the land cover of a region, a feature may be soil type.

Make sure to input all of the files you want to upload at once.

If you plan to re-run this workflow with the same data, you can expand the cell and add the paths to the `value` parameter, which sets a default value. For example, `value = '...data/feature1.tif,...data/feature2.tif'`.

Mask

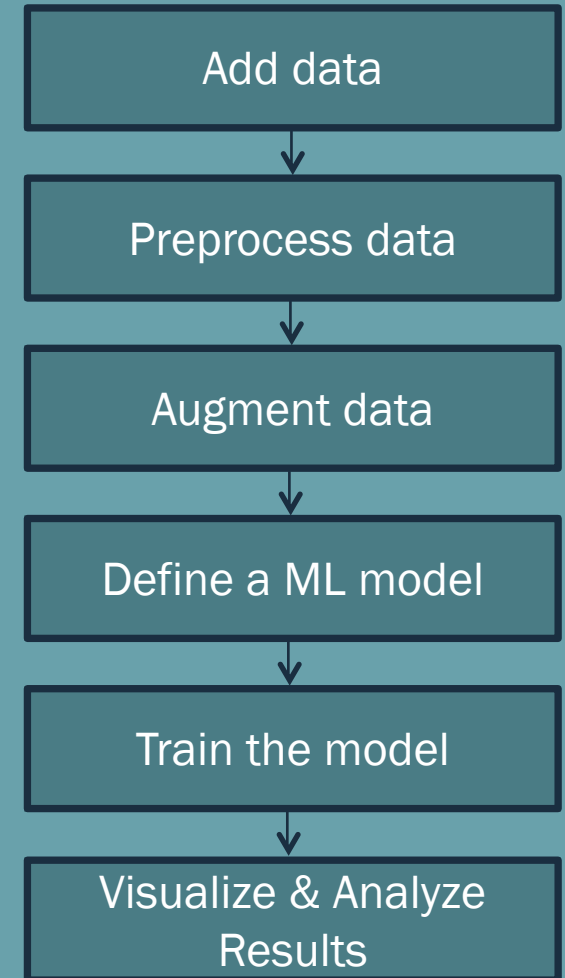
A mask define the output the model learns to predict. For example, if you want to predict the land cover of a region, the masks would be polygons representing the land cover types.

As of right now, this workflow only supports the ability to predict a single mask.

You can only upload a single file.

add the paths to your data files ...

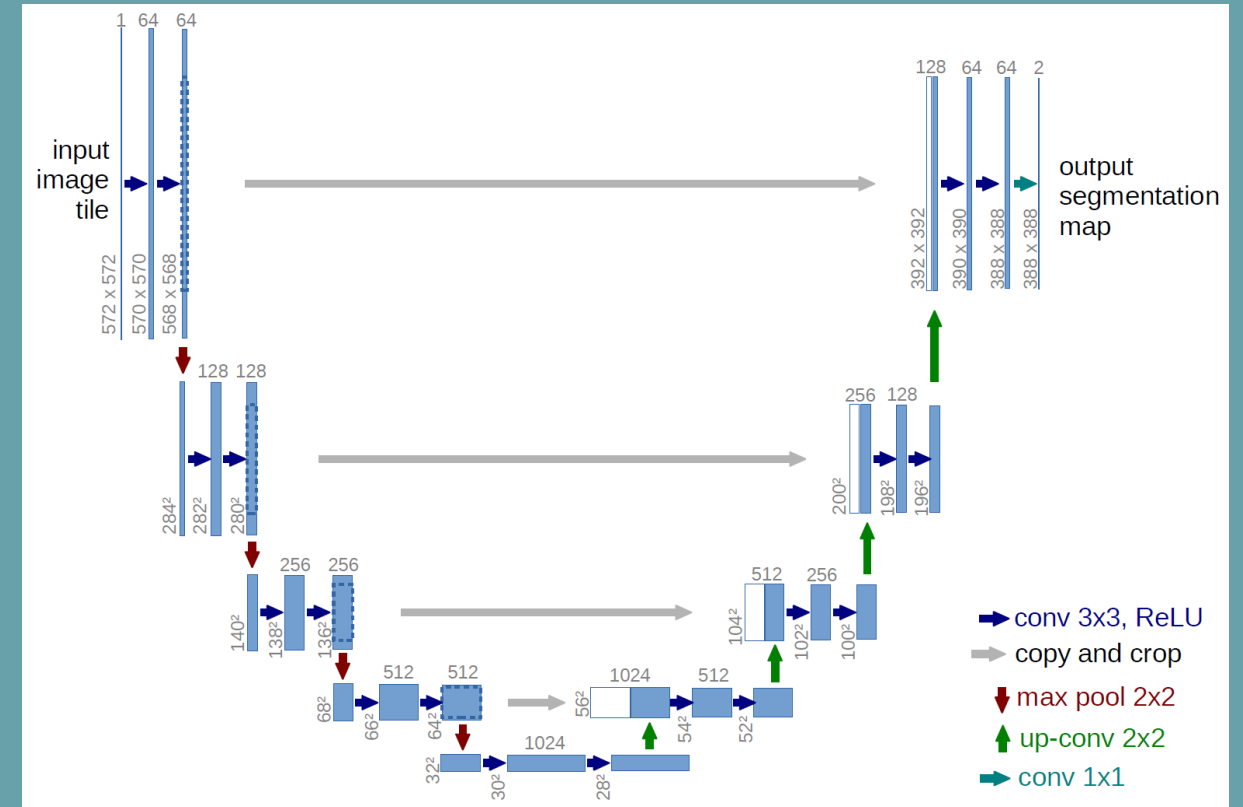
Input Feature Paths	Input Mask Path
Features: <code>../chile_data/alos_nova_friburgo_5m.tif,../chile_data/2328825_2011-08-13_RE1_3A_Analytic.tif</code>	



← This is what it looks like!

The Model: UNet

- The UNet model is a convolutional neural network (CNN) designed for semantic image segmentation, named for its “U” shape
- It makes predictions on a pixel-wise basis
- The model has two paths: down-sampling (left side) and up-sampling (right side)
 - *Down-sampling extracts image features*
 - *Up-sampling localizes objects*
- This kind of neural network is well-suited to geospatial analysis because it can **preserve spatial relationships** in the data



Ronneberger et al. (2015)

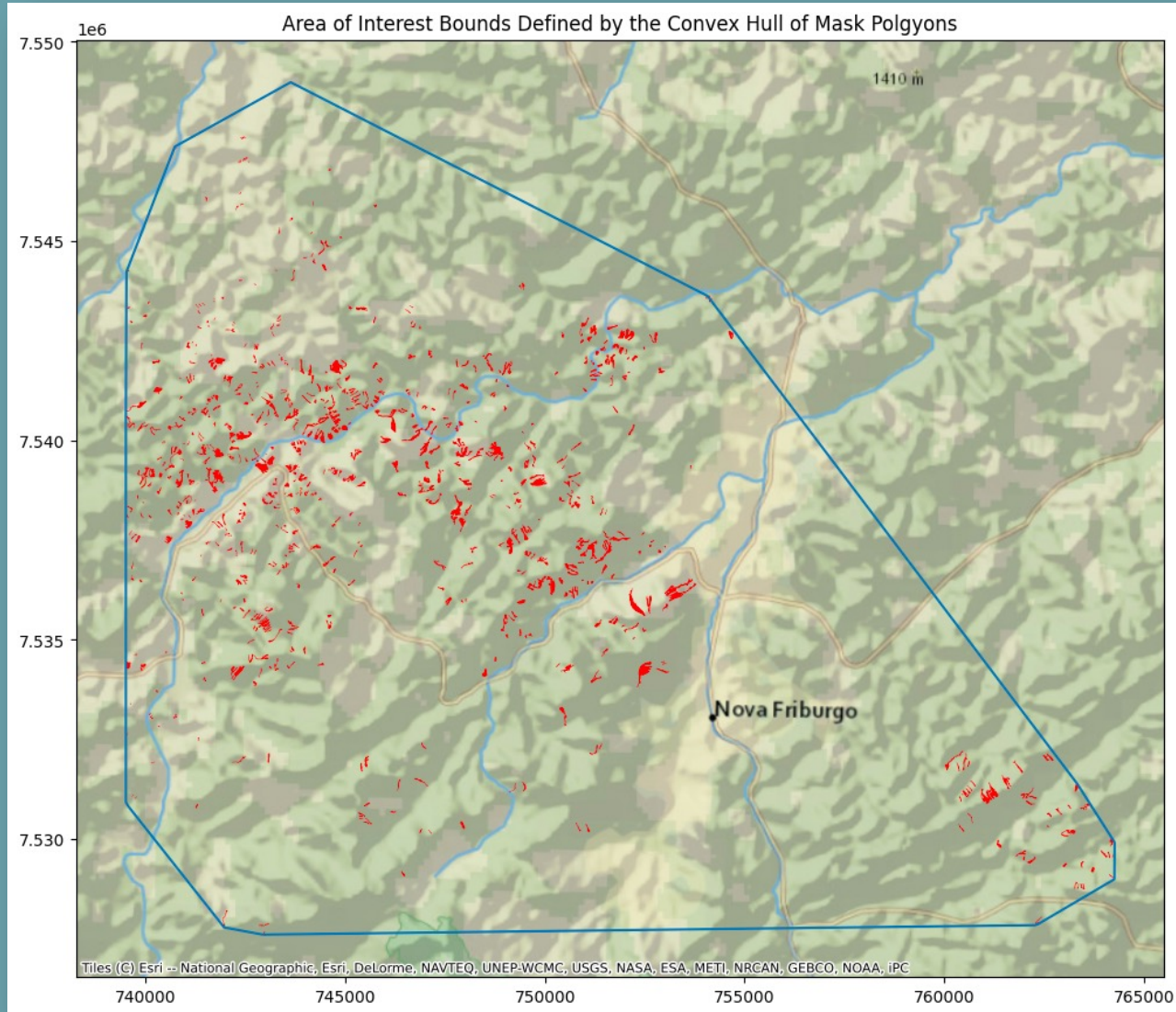


Shi et al. (2021)

A CASE STUDY

Identifying Landslides in Brazil





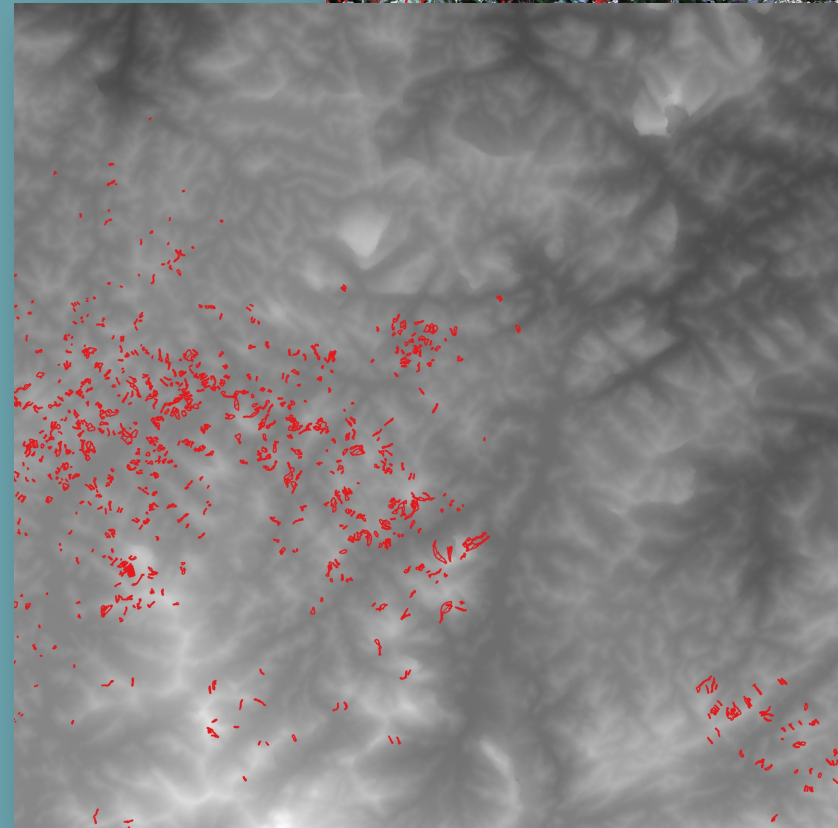
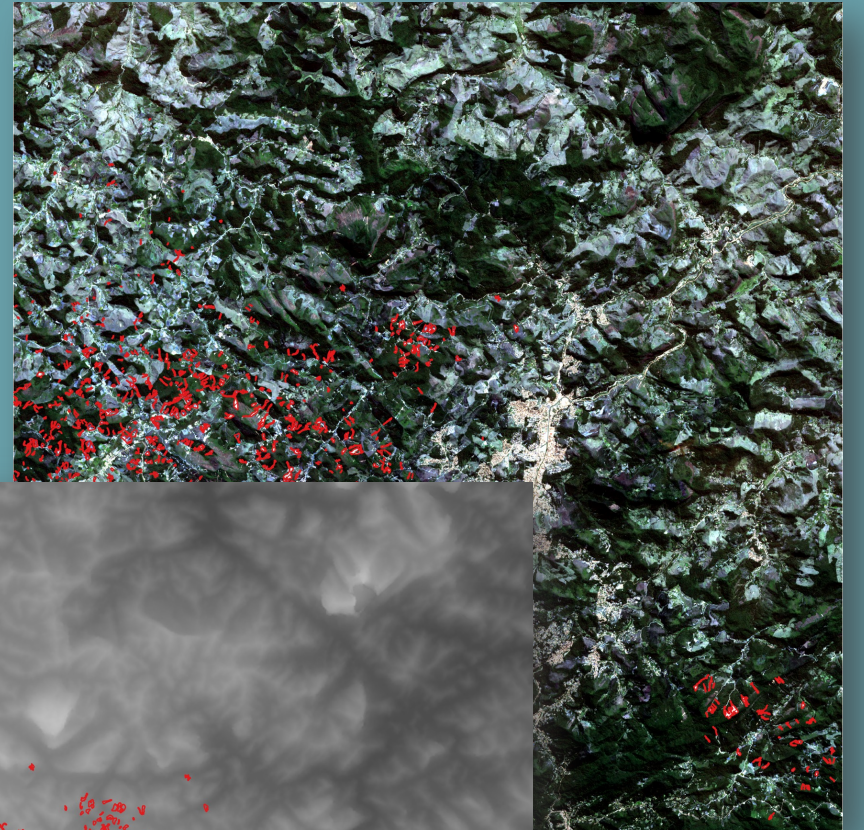
Area of Interest

- The red polygons outline landslide scars (masks)
- The blue polygon defines the area the model will learn from (bounds)
- The bounds are automatically determined by the workflow as the maximum convex polygon around the masks

Input Data

- 5m resolution digital elevation model (DEM)
- RapidEye hyperspectral data (5 bands)
 - *Red, Green, Blue, Red Edge and Near Infrared*

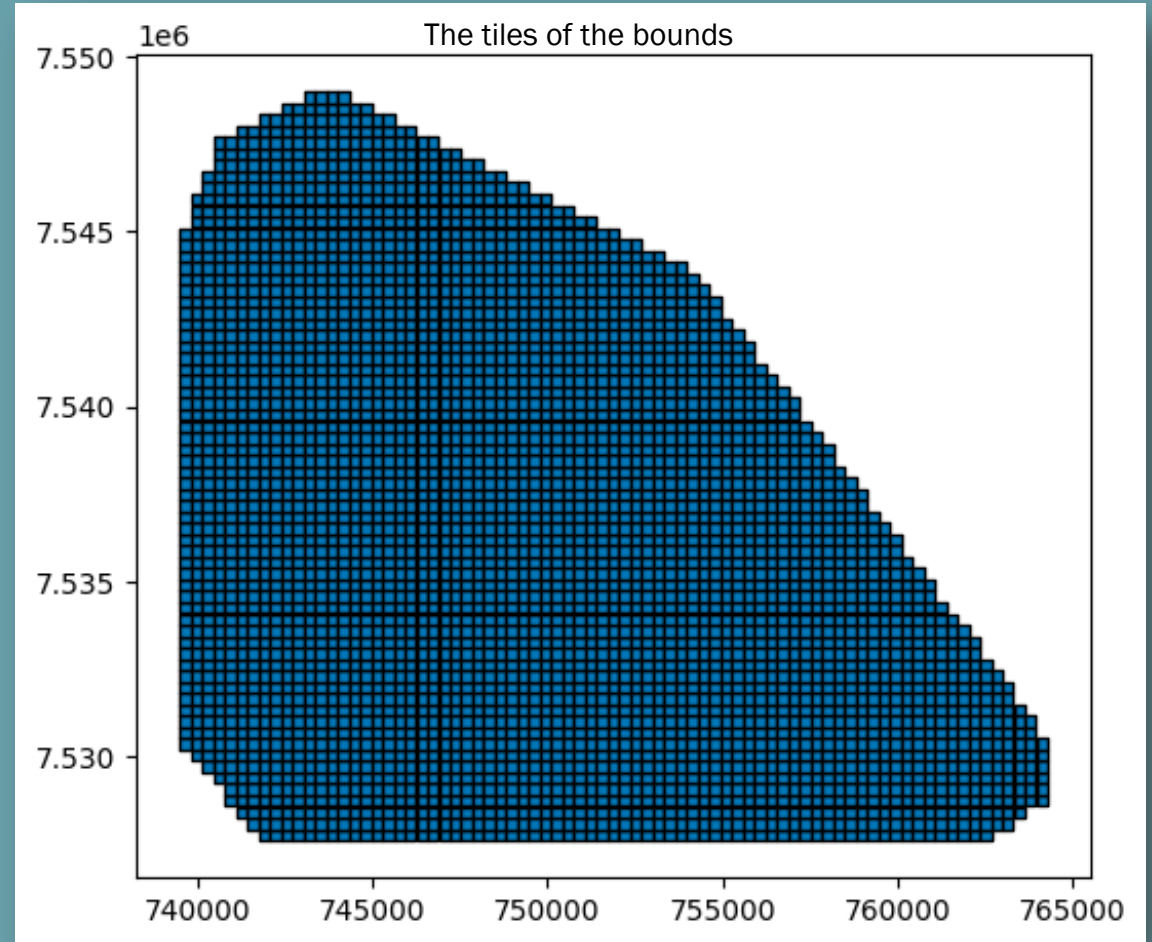
Hyperspectral



DEM

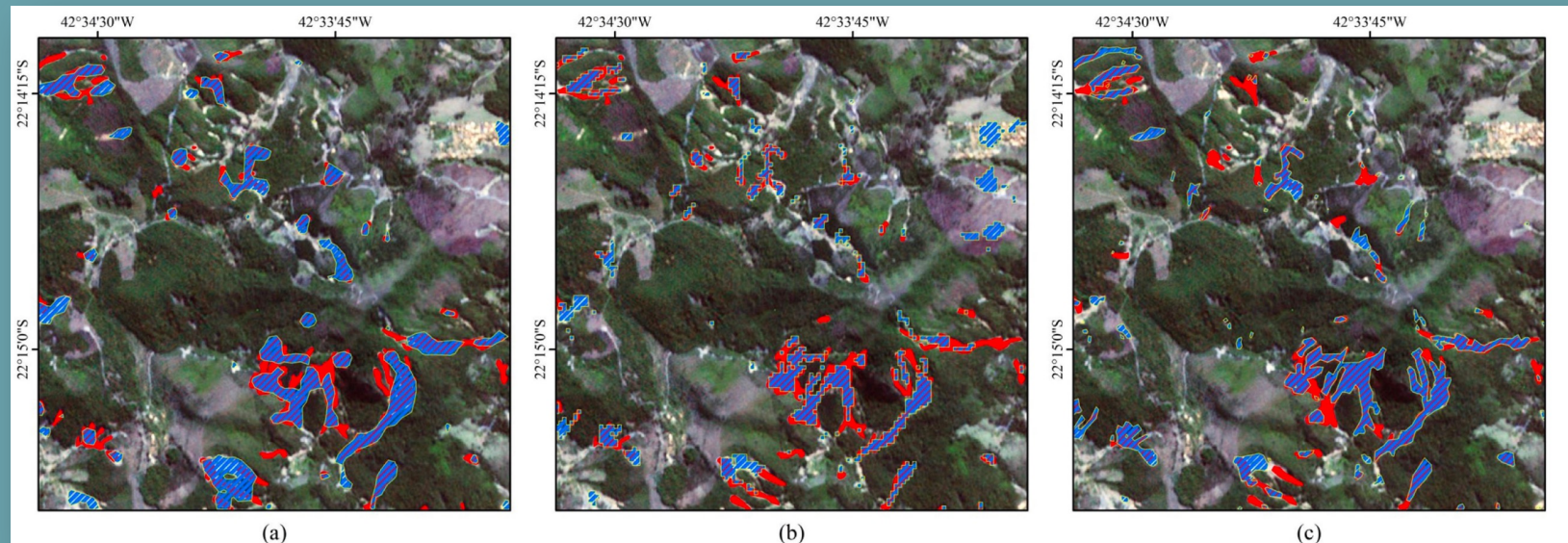
Stacking & Tiling the Data

- To format the data to be used in the model, all of the bands from each of the input data features needs to be **stacked** (composite raster)
 - *This way, the model can learn from all of the data inputs*
- Then, the resulting image needs to be **tiled** into smaller images (tiles)



Results

- While final testing with this dataset has not been completed yet, the results of the model will resemble this figure
- This figure is taken from Xu et al. (2022) who used a similar method to produce predicted masks of the landslides
 - *We aim to benchmark our results against this dataset and others*



Next Steps

- Finalize model parameterization and architecture
 - *Use an updated architecture that can process data with a large class imbalance and few training samples*
- Benchmark results on existing datasets (Brazilian landslide data from Xu et al., etc.)
- Find some beta testers!

References

1. 3D Elevation Program | U.S. Geological Survey. <https://www.usgs.gov/3d-elevation-program>.
2. Wang, L., Wang, C., Sun, Z. & Chen, S. An Improved Dice Loss for Pneumothorax Segmentation by Mining the Information of Negative Areas. *IEEE Access* 8, 167939–167949 (2020).
3. Attention Res-UNet with Guided Decoder for semantic segmentation of brain tumors | Elsevier Enhanced Reader. <https://reader.elsevier.com/reader/sd/pii/S1746809421006741?token=A90DC1240DE9786392A8FC82F56077F03B64E1FCD2A0B3403243E18848CD9C4C76094B666BF58265D291C67F51FAA743&originRegion=us-east-1&originCreation=20230307233744> doi:10.1016/j.bspsc.2021.103077.
4. Amigo, J. M. & Santos, C. Chapter 2.1 - Preprocessing of hyperspectral and multispectral images. in *Data Handling in Science and Technology* (ed. Amigo, J. M.) vol. 32 37–53 (Elsevier, 2019).
5. Li, T. et al. DCNR: deep cube CNN with random forest for hyperspectral image classification. *Multimed Tools Appl* 78, 3411–3433 (2019).
6. Wu, J. F. Effective use of machine learning to empower your research. *THE Campus Learn, Share, Connect* <https://www.timeshighereducation.com/campus/effective-use-machine-learning-empower-your-research> (2022).
7. Xu, G., Wang, Y., Wang, L., Soares, L. P. & Grohmann, C. H. Feature-Based Constraint Deep CNN Method for Mapping Rainfall-Induced Landslides in Remote Regions With Mountainous Terrain: An Application to Brazil. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 15, 2644–2659 (2022).
8. Chegini, T., Li, H.-Y. & Leung, L. R. HyRiver: Hydroclimate Data Retriever. *Journal of Open Source Software* 6, 1–3 (2021).
9. Soliman, A. & Terstriep, J. Keras Spatial: Extending deep learning frameworks for preprocessing and on-the-fly augmentation of geospatial data. in *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery* 69–76 (Association for Computing Machinery, 2019). doi:10.1145/3356471.3365240.
10. Zhou, X.-Y. & Yang, G.-Z. Normalization in Training U-Net for 2-D Biomedical Semantic Segmentation. *IEEE Robotics and Automation Letters* 4, 1792–1799 (2019).
11. Soliman, A. & Terstriep, J. Processing digital elevation data for deep learning models using Keras Spatial. <https://essopenarchive.org/doi/full/10.1002/essoar.10504457.1> (2020) doi:10.1002/essoar.10504457.1.
12. Science, O.-O. D. Properly Setting the Random Seed in ML Experiments. Not as Simple as You Might Imagine. *Medium* <https://odsc.medium.com/properly-setting-the-random-seed-in-ml-experiments-not-as-simple-as-you-might-imagine-219969c84752> (2019).
13. pyproj 3.4.0 documentation. <https://pyproj4.github.io/pyproj/stable/index.html>.
14. Diakogiannis, F. I., Waldner, F., Caccetta, P. & Wu, C. ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data. *ISPRS Journal of Photogrammetry and Remote Sensing* 162, 94–114 (2020).
15. Jadon, S. SemSegLoss: A python package of loss functions for semantic segmentation. *Software Impacts* 9, 100078 (2021).
16. Stewart, A. J. et al. TorchGeo: Deep Learning With Geospatial Data. Preprint at <https://doi.org/10.48550/arXiv.2111.08872> (2022).
17. Ronneberger, O., Fischer, P. & Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015* (eds. Navab, N., Hornegger, J., Wells, W. M. & Frangi, A. F.) vol. 9351 234–241 (Springer International Publishing, 2015).
18. Yeung, M., Sala, E., Schönlieb, C.-B. & Rundo, L. Unified Focal loss: Generalising Dice and cross entropy-based losses to handle class imbalanced medical image segmentation. *Computerized Medical Imaging and Graphics* 95, 102026 (2022).
19. Widget List — Jupyter Widgets 8.0.2 documentation. <https://ipywidgets.readthedocs.io/en/stable/examples/Widget%20List.html#Numeric-widgets>.