Automated mapping of Arctic permafrost tundra using sub-meter resolution satellite imagery

Rajitha Udawalpola
Postdoctoral Scholar

Chandi Witharana
PI, Assistant Professor

Department of Natural Resources and the Environment
University of Connecticut
How monitor **Pan-Arctic** scale permafrost disturbances?)

‘Big’ commercial satellite imagery to rescue

- Entire Arctic (above 60°N) has been imaged by DigitalGlobe Inc. commercial satellites in 0.5m resolution four times in the last 6 years.
- ~ 2 petabytes of imagery (> 1 million images)
- Image data is freely available for NSF-Funded researchers via Polar Geospatial Center (PGC), University of Minnesota
Mapping application for Arctic Permafrost Land Environment - MAPLE

BIG imagery

Pre-processing

Automated classification

Products

PGC – Polar Geospatial Center
XSEDE – eXtreme Science and Engineering Development Environment
Longhorn – Frontera/Texas Advanced Computing Center

Want to run in different computational Setups such as Frontera(GPU/Longhorn), CPU Clusters, Kubernetes
Satellite image data request from User

- Scene Ids
- AOI (shapefile, kml,..)
- ----

Image attributes
- cloud cover
- bands
- sensor
- bit depth
- pansharpening
- product level (ortho)
- stereo/mono
- ----

Bridges Pylon

Polar Geospatial Center (PGC)

[ email to point-of-contact at PGC ]

User account

- Image data
- Analysis
- Codes
- Intermediate Image tiles
- Intermediate Shapefiles
- Final Maps

Model runs

XSEDE

Bridges GPU-AI

PGC does pre-processing and transfer

XSEDE Job Submission by User
Challenges, Bottlenecks, Wish list

Processing of large volumes of jobs on XSEDE

- manual submission
- bash script
- job tracking/management is manual
- submit each image as an individual job
- submit one large job for multiple images?

We are hoping to use PARSL to
- automate job submission
- track and manage jobs
- save the results.
Challenges, Bottlenecks, Wish list

Workflow optimization

One image

Split into ~65,000 tiles

~65,000 vector files

One final map

Image scene

Stage 1: Partition input image into small tiles

Tiling

Stage 2: DL CNN model Infer each and produce shapefiles

Stitching

Stage 3: Merge all the shapefiles

Map

Input

Output

GDAL, numpy, openCV

multiprocessing, queue, shapefile, scikit, tensorflow, numpy, utils, keras, json, scipy

shapefile, GDAL, scipy, numpy
Extra Challenges in the Workflow.

- In some servers there is a conflict between these two. Automated workflow failed in these servers.
- Used two Conda environments for preprocessing and classifications.
- Used two Executors for two steps.
- Could not used object serialization to pass data.
- Used Files to pass data between python Apps.
Acknowledgements for PARSL implementation.

• Daniel S. Katz
• Kyle Chard
• Yadu Babuji