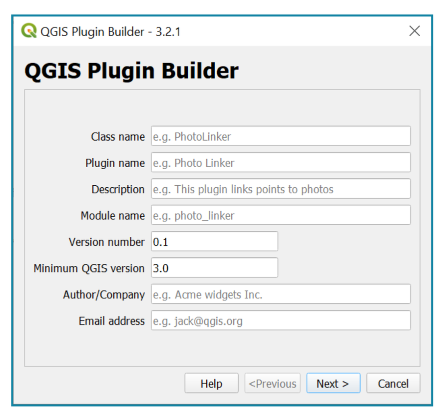
"OpenCloudRemover": A Python-based QGIS Plugin for Efficient Cloud Removal from Sentinel2 Imagery Using Machine Learning

-William Simone -

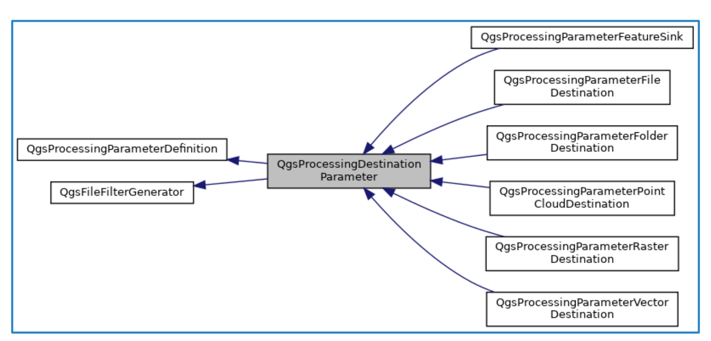
Background

Cloud cover in satellite imagery can lead to inaccuracies and errors when producing maps. To mitigate these issues, it is important to remove the cloud cover prior to analysis and processing. Although Fmask is a commonly used method for cloud removal and has been available since 2012, it may not be suitable for more detailed work, particularly in forested regions of Maine. Therefore, there is a need for new and improved methods that can accurately remove clouds and facilitate analysis of satellite imagery in diverse geographic contexts. This project began in the summer of 2022 to address this issue. During that time researchers from MCC and Umaine collaborated toward the goal of training a ML algorithm (XGBoost) to detect cloud and shadow pixels in satellite imagery acquired from Sentinel-2.
Due to the success of the ML training and application, follow on development was pursued. This is where my work began on the development of a QGIS plugin. The plugin needed to contain all elements of this complex operation in one user friendly package that functions in the QGIS environment.

Methodology



The "OpenCloudRemover" plugin was developed using Python and the QGIS API. To start, QGIS Plugin Builder 3 was utilized to create the initial plugin folder and corresponding .py modules.
Following this a flowchart was designed to layout the package folder structure.



Next research into QGIS API classes found a few tools with the functionality needed but ultimately trial and error led to finding the right functionality and GUI.

I combined these tools with segments from the existing Fmask Python file that processes Sentinel2 data from the command line. Finally, I compiled the required dependencies to ensure the plugin could be distributed and operate independently as intended.

```python
class FmaskTest_Plugin(object):

    def __init__(self):
        self.provider = None

    def initProcessing(self):
        """Init Processing provider for QGIS >= 3.8."""
        self.provider = FmaskTestProvider()
        QgsApplication.processingRegistry().addProvider(self.provider)

    def initGui(self):
        self.initProcessing()

    def unload(self):
        QgsApplication.processingRegistry().removeProvider(self.provider)

```

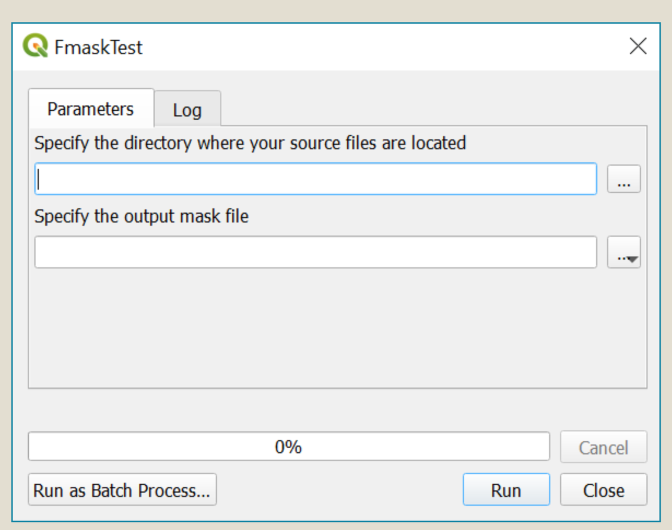


Figure 1. Maiden launch of first test plugin. A bare bones interface and limited user functionality, it left something to be desired. However, an important milestone.
This was created using the QGIS Plugin Builder Plugin. With the provided building blocks additional functionality can be added at the developers pace.

Research Question

Can a trained Machine Learning Algorithm be applied to Sentinel 2 data to provide more efficient high accuracy cloud removal?
Can this process be packaged into a standalone open source plugin for QGIS?

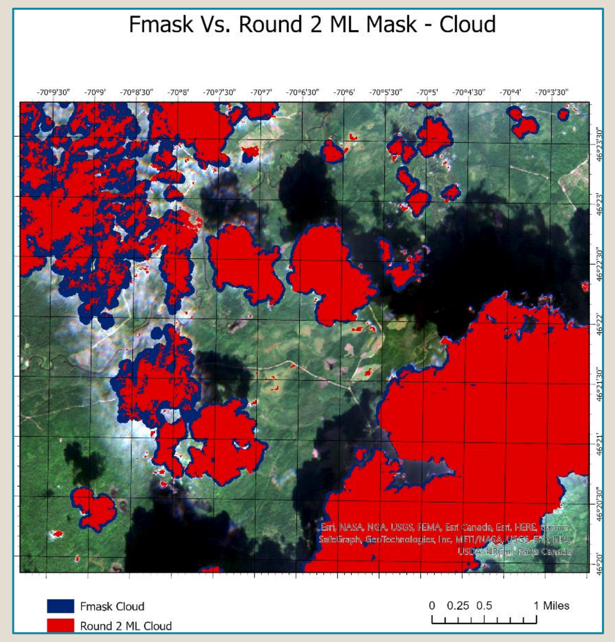


Figure 2. Comparing output for 2 rounds of training for ML algorithm to the current standard, Fmask .

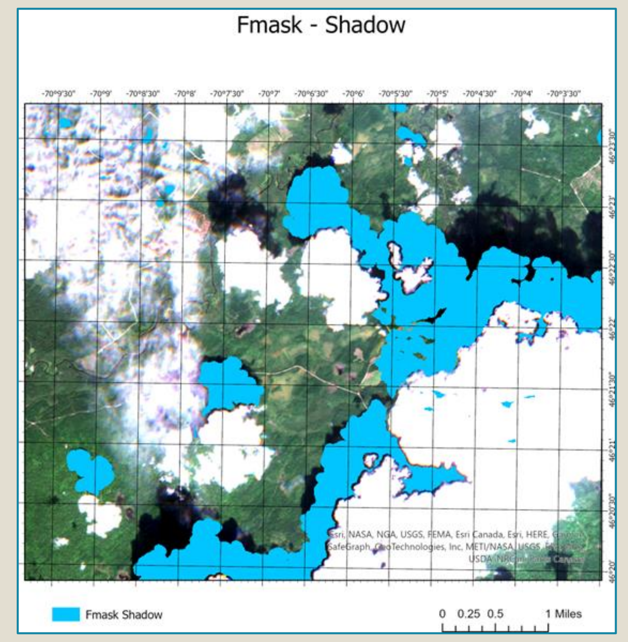


Figure 3. Glaring errors with current standard on shadow masking. Missed shadows and unfilled holes in cloud mask.

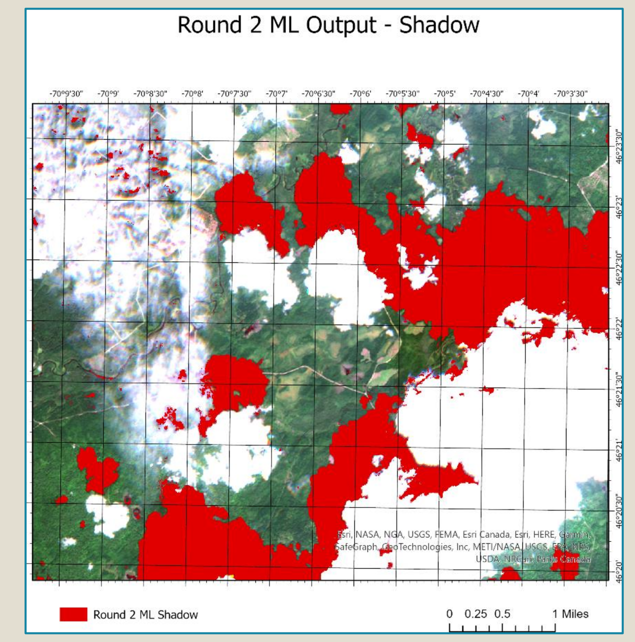


Figure 4. 2 rounds of training ML algorithm results in high shadow detection and better, more detailed coverage.

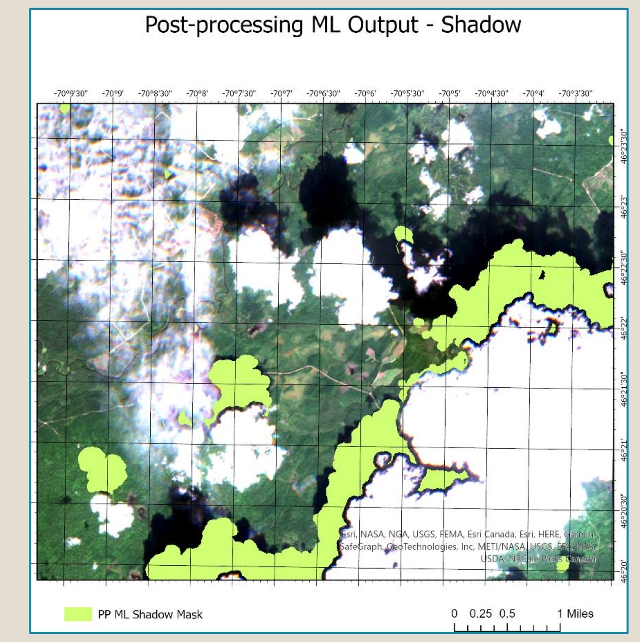


Figure 5. Post-processing results from Round 2 ML output diminishes returns, creates less accurate result.

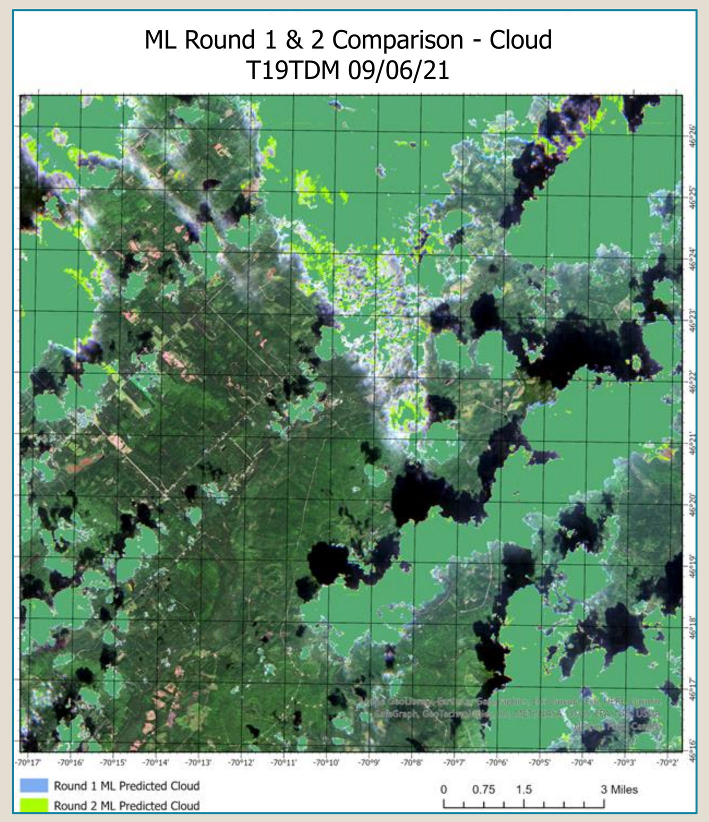


Figure 6. Both Rounds of trained ML output show little variation in cloud detection.

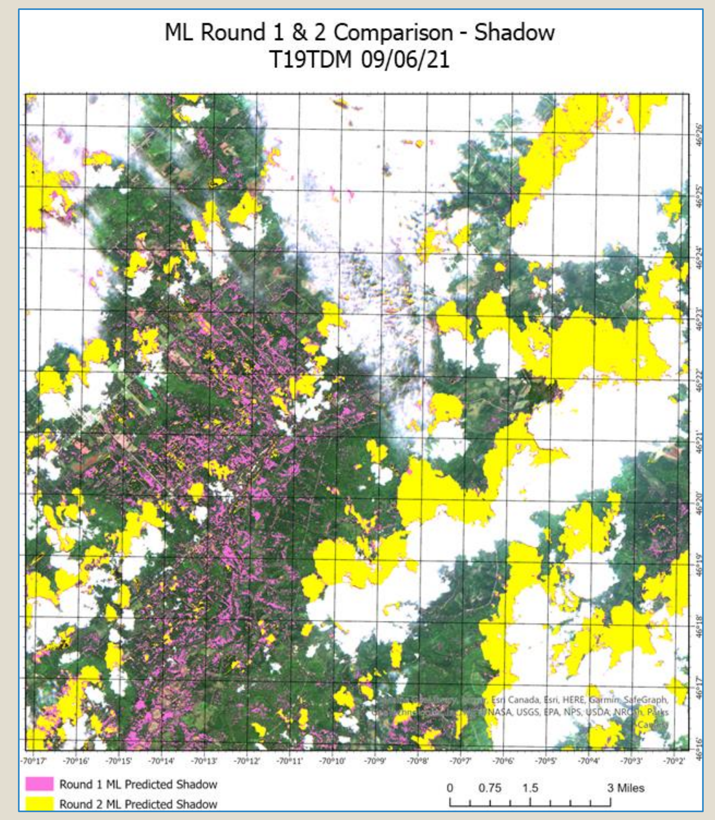


Figure 7. Both Rounds of trained ML output placed together show increased accuracy of identifying shadow pixels with a second round of training.

ML Training Results

Fmask VS. Round 2 ML Output – Clouds **(Figure 2)**
- Fmask cloud coverage is greater, but less detailed (due to buffering of mask)
- Round 2 ML output underestimates cloud coverage

Shadow Output **(Figures 3 to 5)**
- Fmask overestimates areas of shadow in areas of visible clear pixels
  - Underestimation of shadow in areas of visible shadow pixels
- Round 2 ML output refines shadow coverage, reduces overestimation from Round 1
- Post-processing ML Fmask output is less accurate than the Round 2 ML output

ML Round Comparison **(Figures 6 & 7)**
- Cloud matching nearly identical, increased thin cloud detection with Round 2
- Shadow refinement reduces errors, fewer clear pixels labeled as shadow

Post-processing ML Fmask output VS. Round 2 ML Output **(Figure 8)**
- PP ML predicts clouds with same high accuracy as Round 2 ML output, but adds a buffer
  - Beneficial in areas of wispier, thin clouds
- Round 2 predicts shadow much more accurately than PP ML output.
  - PP ML still relies on cloud-shadow matching. Highly selective algorithm needs to be fine tuned to allow for more variation in potential best-match imagery.

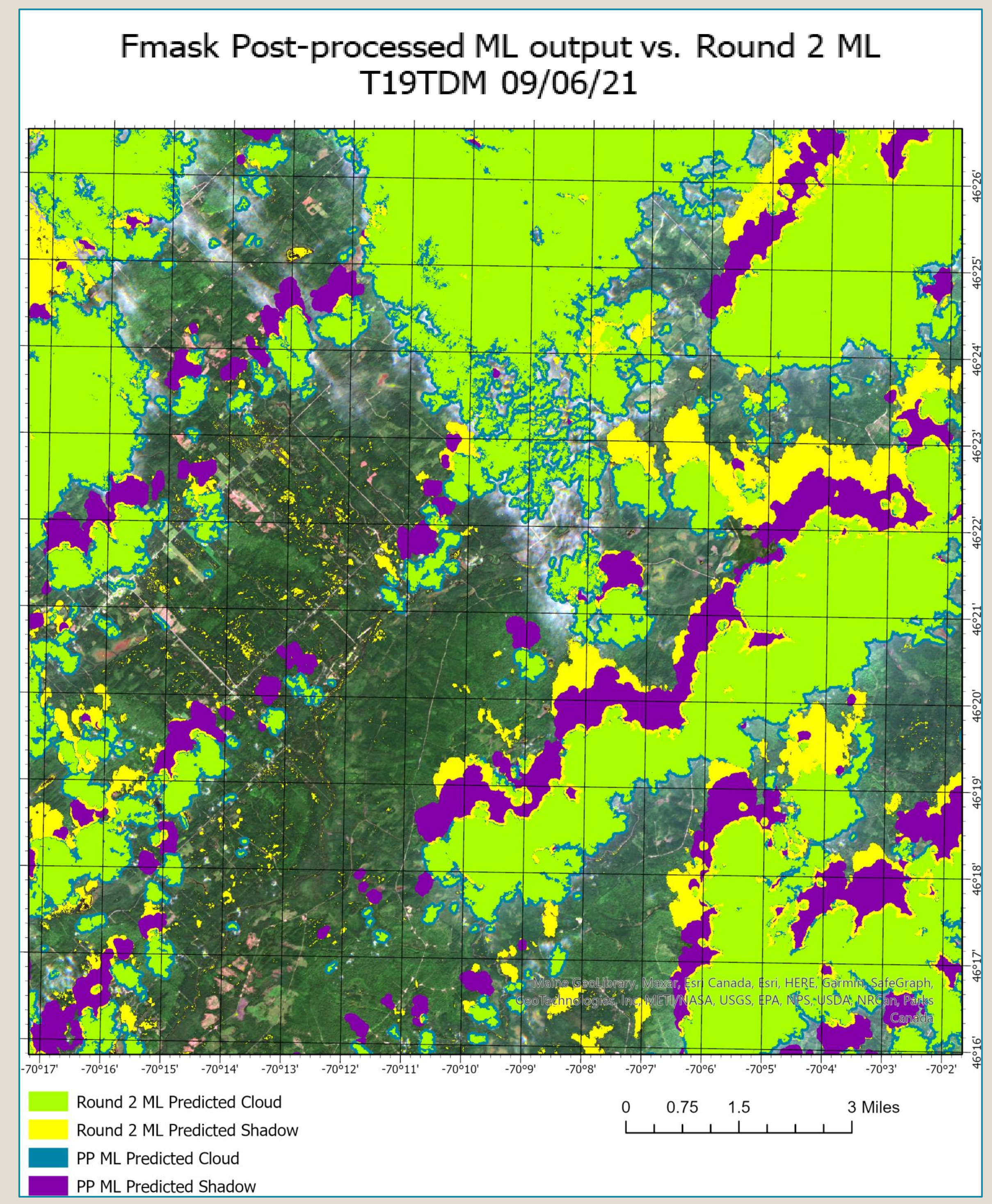


Figure 8. Round 2 ML prediction masks with Post-processed ML Fmask output. Optimal combination of layers to have the most cloud/shadow coverage would be either ML or PP ML cloud mask, paired with the Round 2 ML predicted shadow.

Fmask shadow prediction needs to be altered to allow for greater variation in where potential shadow layer is, and refinement of best-match properties.

Further rounds of training could enhance cloud detection, and reduce further shadow overestimates on areas of clear pixels.

*Images acquired from Sentinel-2.
All maps are projected in WGS 1984 UTM Zone 19N.

Future Work

- Perform accuracy assessment & quantitative comparison on ML output
  - Compare to Fmask accuracy (between 92.4 and 96.4%, dependent on version and source of imagery)
- Further refinement of ML
  - Include atmospheric correction
  - More training points for shadow layer
- Train for cloud/shadow detection on fall/spring imagery
- QGIS Plugin
  - Enhance user interface
  - Allow for multithread processing in the QGIS environment
  - Multi satellite functionality

References & Acknowledgements

University of Maine – Center for Research on Sustainable Forests, School of Forest Resources
Dr. Kasey Legaard, *Assistant Research Professor*
Barbara Wheatland Geospatial Lab
Dr. Daniel Hayes, *Director*
Anthony Guay, *Remote Sensing Specialist*
David Sandilands, *Aerial Survey Pilot & Remote Sensing Technician*
University of Maine – Center for Advanced Forestry Systems
Dr. Aaron Weiskittel, *Director*
Monroe Community College – Department of Chemistry & Geosciences
Jonathon Little, *GIS Professor & Principal Investigator*
Wayne Howard, *Computer Science & GIS Professor & Senior Personnel*
Casmir Brown, *GIST Alum and Researcher*

- Miller, Michael. "QGIS Plugin Development with Python." Udemy, Udemy Inc., 2020, www.udemy.com/course/qgis-plugin-development-with-python/.
- Zhu, Z., & Woodcock, C. E. (2012). Object-based cloud and cloud shadow detection in Landsat imagery. Remote Sensing of Environment, 118, 83–94. https://doi.org/10.1016/j.rse.2011.10.028
- Zhu, Z., Wang, S., & Woodcock, C. E. (2015). Improvement and expansion of the Fmask algorithm: cloud, cloud shadow, and snow detection for Landsats 4–7, 8, and Sentinel 2 images. Remote Sensing of Environment, 159, 269–277. https://doi.org/10.1016/j.rse.2014.12.014

This experience is thanks to Skills Training in Advanced Research & Technology (START) Supplemental Funding Request for ATE at Monroe Community College (Award #1955256) with IUCRC Phase 3 at University of Maine - Center for Advanced Forestry Systems (CAFS). Available for educational use only. Created 2022.