# Operational-Scale GeoAI for Pan-Arctic Permafrost Feature Detection from High-Resolution Satellite Imagery

Mahendara Udawalpola<sup>1\*</sup>, Amit Hasan<sup>1</sup>, Anna K. Liljedahl<sup>2</sup>, Aiman Soliman<sup>3</sup>, Chandi Witharana<sup>1</sup>

<sup>1</sup>Dept. of Natural Resources and the Environment, University of Connecticut, USA – (\*mahendra.udawalpola, amit.hasan, chandi.witharana)@uconn.edu

<sup>2</sup>Woodwell Climate Research Center, Falmouth, Massachusetts, USA – aliljedahl@whrc.org

<sup>3</sup>National Center for Supercomputing Applications, University of Illinois Urbana-Champaign, USA – asoliman@illinois.edu

KEY WORDS: Arctic, Permafrost, Ice-wedge polygons, Deep learning, Satellite imagery, High Performance Computing

#### ABSTRACT:

Regional extent and spatiotemporal dynamics of Arctic permafrost disturbances remain poorly quantified. High spatial resolution commercial satellite imagery enables transformational opportunities to observe, map, and document the micro-topographic transitions occurring in Arctic polygonal tundra at multiple spatial and temporal frequencies. The entire Arctic has been imaged at 0.5 m or finer resolution by commercial satellite sensors. The imagery is still largely underutilized, and value-added Arctic science products are rare. Knowledge discovery through artificial intelligence (AI), big imagery, high performance computing (HPC) resources is just starting to be realized in Arctic science. Large-scale deployment of petabyte-scale imagery resources requires sophisticated computational approaches to automated image interpretation coupled with efficient use of HPC resources. In addition to semantic complexities, multitude factors that are inherent to sub-meter resolution satellite imagery, such as file size, dimensions, spectral channels, overlaps, spatial references, and imaging conditions challenge the direct translation of AI-based approaches from computer vision applications. Memory limitations of Graphical Processing Units necessitates the partitioning of an input satellite imagery into manageable sub-arrays, followed by parallel predictions and post-processing to reconstruct the results corresponding to input image dimensions and spatial reference. We have developed a novel high performance image analysis framework – Mapping application for Arctic Permafrost Land Environment (MAPLE) that enables the integration of operational-scale GeoAI capabilities into Arctic science applications. We have designed the MAPLE workflow to become interoperable across HPC architectures while utilizing the optimal use of computing resources.

#### 1. INTRODUCTION

Big image data analysis has become essential in an array of scientific applications, such as computer vision (CV) (Kucuk et al., 2017), medical imaging (El-Baz and Suri, 2019), material science (Okunev et al., 2020), astronomy (Kremer et al., 2017). Owing to the advancements of satellite sensor technology coupled with ever increasing spatial resolution and temporal frequency of image acquisitions ideally position Remote Sensing applications in a 'big' data landscape. Satellite imagery archives are radically being transformed from terabytes to petabyte scale. Sheer volumes of imagery pose new challenges in storage, analysis, and visualization techniques. These requirements exceed the capabilities existing general purpose computing resources. The quest is therefore at its peak for seamless integration of high-performance computing (HPC) resources to translate big satellite imagery into science-ready products, which enable knowledge discovery at the nexus of the human-natural system.

In recent years, usage of HPC resources has become an inextricable component in big imagery applications. Highly efficient workflows are indeed required for automatically analyzing hundreds-to-thousands of satellite imagery. Traditional RS image analysis algorithms fail to grapple with the image complexities and high-level semantics arising from the sub-meter resolution satellite imagery. Sophisticated algorithms, which exploit color, texture, spatial arrangement, and context and construct high-level abstractions based on low-level motifs are needed for automated object detection, segmentation, and classification. Deep learning (DL) convolutional neural nets (CNNs) (LeCun et al 2016) has shown great potential for object instance segmentation in detecting and delineating each distinct

object in an image of common objects from everyday images. The success of DLCNNs in CV applications has received great interest from the remote sensing community. DLCNNs algorithms are computationally intense and memory demanding. Thus, it is important to optimize data management, image processing, classification, and visualization techniques as they can be bottleneck in image-to-assessment pipelines.

There are many applications found in literature involving big imagery and HPC computing. (Amat et al., 2015) have developed a workflow for light-sheet microscopy, which involves several tens of terabytes of data. Schmied et al. (2016) have compared the performance of an automated workflow between a single workstation and HPC cluster. Liu et al. (2016) have analysed geosciences workflow on multi-core processors and Graphical Processing Units (GPUs). They have achieved 5.x speed up on a multi-core processor and 43.x speedup for some parts of the workflow on GPU. A-Saadi et al. (2020) compared workflow application designs for high resolution satellite imagery analysis. They have analysed three workflows using Extreme Science and Engineering Discovery Environment (XSEDE) HPC computer for two use cases with 4672 images and 8.35 TB of data.

Modern HPC systems consist of a large number of HPC computer nodes. Each node contains multi-core Central Processing Units (CPUs) and multi-GPUs. These resources measure the usage using different accounting models. Different configurations present new challenges to design efficient workflows for different applications, which require both CPU and GPU processing.

Many efforts have been made to identify changes in the environment using geospatial big data. Archived observation data was predicted to exceed Exabyte by 2015 by the Open Geospatial Consortium (OGC). But it is estimated that upto 95% of the data present in existing archives has never been accessed. (Nikolaouet al. 2014)

The entire Arctic has been imaged at 0.5 m resolution by commercial satellite sensors (DigitalGlobe, Inc.). The image repository at the Polar Geospatial Center (PGC) provides transformational opportunities to observe, monitor, and document permafrost thaw occurring across the Arctic, which is a remote region with an extremely sparse field observation network. Alaska, Canada, and Russia collectively harbor approximately 5 million km<sup>2</sup> of tundra. But, yet imagery-derived science products are rare despite their unprecedented potential for Arctic science applications.

Arctic permafrost - unique landscapes comprising the Earth materials that remain at or below 0°C for at least two consecutive years - occupies approximately 24% of the exposed land surface of the northern hemisphere. Ice-rich permafrost can be identified by atypical surface features called ice-wedge polygons (IWPs), which are underlain by several meter-wide and deep ice-wedges that form a network across the tundra. Vegetation and geology maps suggest that about two-thirds or more of the Arctic landscape is occupied by polygonal ground (Raynolds et al. 2019) and therefore ice-rich ground, but the exact extent and the prevailing IWP types (i.e., whether the ice wedges experienced melt or not) are largely unknown (Liljedahl et al. 2016).

Over recent decades, ice-wedge degradation - transformation of low-centered polygons into high-centered polygons due to icewedge degradation has been documented at several locations across the Arctic tundra in the field and through localized remote sensing analyses (Steedman et al., 2017). The shift from one IWP type to the other is documented to occur in less than a decade (Liljedahl et al. 2016) with an unusual warm summer, wildfires, or human activities initiating the onset of ice-wedge degradation. Degradation of ice wedges is a quasi-cyclic process with degradation often occurring over a shorter time scale than the formation of new permafrost (aggradation), with the latter controlled by accumulation of organic and mineral soil above the ice-wedge (Kanevskiy et al. 2017). Understanding of spatiotemporal dynamics behind the evolution of ice-wedge polygonal tundra demands for objective and detailed maps consolidating the ice wedge extent and their prevailing successional stages.

Despite the alarming signals, yet the Arctic science community has a limited understanding of the spatiotemporal continuity of the otherwise locally observed changes. The lack of knowledge on the larger geographical extent and successional stage of IWPs introduce uncertainties to regional and pan-Arctic estimates of carbon, water, and energy fluxes. Remote sensing provides transformational opportunities to observe, monitor, and measure the Arctic polygonal landscape at multiple spatial scales and varying temporal windows. IWPs are difficult to detect in any remote sensing imagery with spatial resolution greater than 4 m. Therefore, sub-meter resolution commercial satellite imagery (e.g. DigitalGlobe, Inc.) demonstrates a greater promise in accurate delineation and characterization of ice-wedge polygonal networks. Due to IWPs' varying spectral and morphometric characteristics, visual inspection and manual digitization has so far been the most widely adopted and promising method to delineate polygons from high resolution remote sensing imagery. A considerable number of local-scale studies have analyzed ice wedge degradation processes using satellite imagery, and manned-/unmanned aerial imagery/LiDAR data (Muster et al. 2013). Most of the studies to date have relied on manual image interpretation and/or semi-automated approaches (Skurikhin et al. 2014). Therefore, there is a need and an opportunity for utilization of VHSR imagery in regional scale mapping efforts to spatio-temporally document microtopographic changes due to thawing ice-rich permafrost.

The goal of our ongoing effort is the production of the first pan-Arctic ice-wedge polygon map using a large volume of commercial satellite imagery available at the Polar Geospatial Center and HPC resources. At the first stage we create an IWP map with high IWP probability regions of the Arctic. This area includes around 25 000 satellite images with 200 TB data. We have developed Mapping Application for Arctic Permafrost Land Environment (MAPLE) workflow, which can be deployed in heterogeneous computing resources. Main objective of this paper is to analyse the computational efficiency of the workflow in heterogeneous computational environments, which involves both CPUs and GPUs.

#### 2. METHODS

# 2.1 Mapping Application for Permafrost Land Environment

A workflow diagram of the MAPLE is shown in Figure 1. The main objective of the workflow is to detect ice-wedge polygons in the entire Arctic permafrost tundra region. The workflow should process approximately 5 million km<sup>2</sup> in Alaska, Canada, and Russia, collectively. Further the analysis should be carried out over periodically to detect temporal changes of ice-wedge polygons.

The MAPLE workflow takes high resolution satellite images as input and outputs an ice-wedge polygon map and a map of surface water bodies. The spatial resolution of satellite imagery n is 0.5 m with a typical footprint of 20 km X 20 km (i.e. 160 million pixels per image). The scale of workflow requires sophisticated computation approaches to process images with efficient use of high-performance computing resources.



Figure 1. Simplified schematic of the Mapping Application for Permafrost Land Environment (MAPLE). Imagery © 2016 DigitalGlobe, Inc

The HPC resources are limited and must be shared with other users. Therefore, we need to use the HPC resources efficiently to process images within a given HPC allocation. We have employed several techniques to improve the efficiency of the workflow.



Figure 2. (a) The sample satellite images with water bodies. (b) The calculated water mask by the workflow where water bodies

are represented by black colour., Imagery @ 2016 DigitalGlobe, Inc

The first approach is to minimize the area of processing by removing unwanted areas such as large numbers of waterbodies in the arctic regions. First, water bodies are detected accurately using the methods developed by Kaiser et al. (2019) for mapping lake dynamics. Figure 2 shows the generated binary water mask of a smaller satellite image. The binary mask of generated water bodies is applied on the satellite image to remove water areas from processing. Figure 3 shows the zoomed version of the satellite image on the left and the detected water bodies on the right. We were able to detect small water bodies in our workflow. These calculated water bodies will be used to generate a map of small water bodies in the region.



Figure 3. (a) A zoomed sample satellite image with water bodies. (b) The zoomed calculated water bodies by the workflow where water bodies are represented by black. Imagery © 2016 DigitalGlobe, Inc

The satellite footprint has considerable amount of image overlap due to different imaging times and different sensors. We can considerably reduce the processing volume by removing these image overlaps. We have developed an algorithm which calculates the image overlaps and excludes them from processing in our workflow.

We use Mask RCNN (He et al. 2016) as the key DLCNN model in MAPLE. These DLCNN models perform much better in GPU than in CPU. Usually, the amount of memory available in GPUs is much smaller than that in CPUs. Therefore, we cannot fit a complete satellite image together with the DLCNN model in GPU memory. Due to this requirement we need to split the satellite image into small patches. These patches are saved using a compressed file format after excluding water bodies and nodata areas of the image. These patches are accessed in parallel within the workflow and outputs detected ice-wedge polygons. Each parallel process stores polygons in an individual shapefile.



Figure 4. (a) The original satellite image with ice-wedge polygons. (b) Plot of the ice-wedge polygonsImagery © 2016 DigitalGlobe, Inc

The Figure 4 shows the map of ice-wedge polygons from the MAPLE workflow. The number of polygons detected in an image vary a lot with the image quality, tundra type, landscape.

We keep 10% overlap between patches not to miss polygons intersected by patch boundaries. This results in duplicate polygons along the patch edges. We combine the shapefiles generated by each parallel process into a single shapefile and remove duplicate polygons during the post-processing stage. The Figure 5 shows a zoomed view of a section of a satellite image on the left and the detected polygons by the model on the right.



Figure 5. (a) The zoomed satellite image with ice-wedge polygons. (b) Plot of the ice-wedge polygons of the zoomed image. Imagery © 2016 DigitalGlobe, Inc

## 2.2 Model Training

We employ a transfer learning strategy to re-train the Mask-RCNN network. Annotated ice-wedge polygon dataset was created using an online web tool "VGG Image Annotator " from satellite imagery comprising heterogeneous tundra types. We randomly selected 512 cropped subsets from different tundra types (tussock, non-tussock, and sedge) considering the spectral, and spatial variability. The training data set consists of 9200 hand annotated ice-wedge polygons. We started with pre-trained weights generated by the COCO dataset and trained only the head layers of the Mask-RCNN network. Training was implemented using NVIDIA GeForce RTX 2080 GPU with 10 GB memory. We trained the Mask R-CNN model with a mini-batch size of two image tiles, 250 steps per epoch, learning rate of 0.001, learning momentum of 0.9, weight decay of 0.0001 and with 50 epochs.

#### 2.3 Workflow Models

The modern HPC resources, such as Frontera (Texas Advanced Computing Center) and XSEDE consist of multiple nodes. Each node contains multiple CPUs and GPUs. Each CPU and GPU contain multiple cores. The programs should be designed to use these resources optimally. The TensorFlow library used in the Mask RCNN model can run on GPU.

Figure 6 shows the semantic diagram of the sequential workflow in a single computing node. In this setup we do not use multiple CPUs and GPUs available in the node. All three stages preprocessing, inference and postprocessing executed sequentially.



Figure 6. A semantic diagram of single-CPU single-GPU MAPLE workflow model using a single HPC node.

Figure 7 shows the second model where multiple GPUs available in a single computing node are used. Here the patches generated in the pre-processing stage, that is stored in a multi-threaded queue, are processed using multiple GPUs.



Figure 7. A semantic diagram of single-CPU multi-GPU MAPLE workflow model using a single HPC node.

Figure 8 shows a multi-CPU-GPU model where we process multiple images per batch. In the preprocessing we use multiple CPU cores found in a single node. Then tiles from each image are processed in separate nodes for inferencing stage with using optimum number of GPUs available in that node. The shapefiles generated in the inferencing stage are processed in a single HPC node using multi-CPU cores at the post-processing stage.



Figure 8. A semantic diagram of multi-CPU-GPU MAPLE workflow model using multiple HPC nodes.

#### 2.4 Numerical Experiments.

The numerical experiments were carried out in the Frontera HPC system computing nodes described in Table 1. Each node consisted of 2 IBM Power 9 CPUs and each having 20 cores. Each core consisted of 4 hardware threads. The total amount of RAM available was 300 Gb. Each node consisted of 4 Nvidia Tesla V100 GPUs with each having 16Gb memory. There are 96 such computing nodes in the Frontera Longhorn HPC system.

Model	IBM Power System AC922
Processor:	IBM Power 9
Total processors per node:	2
Total cores per processor:	20
Total cores per node:	40
Hardware threads per core:	4
Hardware threads per node:	160
Clock rate:	2.3GHz
RAM:	256GB
GPUs:	4x NVIDIA Tesla V100
GPU RAM:	4x 16GB (64 GB total)

Table 01: Frontera Longhorn computing node configuration.

The effective use of HPC resources depends on the accounting model and HPC architecture. We have shown the optimum set up for the Frontera Longhorn system where job accounting is based on node time. The project resources are allocated based on service units. One service unit (1 SU) is calculated by multiplying job duration in wall clock hours, charge rate per node hour and number of nodes per job. Therefore, to get maximum resource utilization we need to use all four GPUs per job. But in some HPC systems like XSEDE Bridges2 one SU is defined by multiplying job duration, number of GPUs per node, charge rate per one hour and number of nodes. Here we must calculate the optimum number of GPUs for a single job.

#### 3. RESULTS AND DISCUSSION

We evaluated the time taken for three stages of the sequential workflow described in Figure 6 for different images sizes as a base case. Figure 9 shows the computation results. Blue colour denotes pre-processing time, orange denotes inferencing time and colour Gray denotes post processing time. The first bar shows the time taken for a 400-million-pixel image in the CPU. The rest of the bars represent time taken using GPU for inferencing stage for different image sizes. When comparing the first two bars, it is evident that using GPU for inferencing achieves 6.0x speed up. The reason for this speed up is that DLCNN computations can be simultaneous with a large number of GPU cores. When increasing the size of the image the time increases. The time taken to process a 3600-million-pixel image on GPU is in the same order as the time taken to process 400 million pixels using only CPUs.



Figure 9. Comparison of time taken for MAPLE workflow for different image size for CPUs and GPUs.

Most of the HPC nodes are having multiple GPUs in a single node. Therefore, in the model illustrated in Figure 7 we utilize the multiple GPUs to process images. Figure 10 shows the improvement we obtained for the inferencing stage. Using 4 GPUs we can obtain 3.6x speed up for a 160-million-pixel image. We cannot obtain 4.0x speed up due to I/O operations and serial sections in calculation. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLIV-M-3-2021 ASPRS 2021 Annual Conference, 29 March–2 April 2021, virtual



Figure 10. Comparison of time taken for multiple GPUs to process 1600-million-pixel images in a single HPC node.

Figure 11 shows the computation results for full workflow for an image with 1600-million-pixel image. The speed up achieved using 4 GPUs is 2.0x. The reason for lower seep up is due the increase of percentage of serial workload.



Figure 11. Comparison of time taken for multiple GPUs to process 1600-million-pixel images in a single HPC node.

The Figure 12 shows the time taken to perform preprocessing and postprocessing in multiple CPU cores in a single node. The speed up using 10 CPU cores in a single computing node for preprocessing is 7.5x and for post processing is 9.7x which results in a combined speed up of 8.4x. Pre-processing is a memory intensive task. It needs four times the memory of the image during pre-processing. With 256 GB RAM available in one node, we can only process up to ten 1600-million-pixel images.



Figure 12. Comparison of time taken for multiple GPUs to process a 1600-million-pixel image in a single HPC node.



Figure 13. Comparison of time taken for multiple GPUs to process a 1600-million-pixel image in a single HPC node.

Post-processing is less memory intensive. Therefore, we can process more images with post processing. The Figure 13 shows the time taken for post-processing with respect to the number of CPU cores. After 10 CPUs the speed up started to saturate. This is mainly due to I/O operations.

The Figure 14 shows the total amount of time taken to process one 1600-million-pixel image with model 3 with 4 GPUs. The first bar shows the result we obtained with model 2. The speed up of 2.4x with 5 CPUs and 2.9x achieved compared to Model 2. When compared with model 1 a speed up of 3.4x with 5 CPUs and 4.0x with 10 CPUs are achieved. This 4.0x speed up means we can process 4 times faster than sequential mode in Figure 6 with the same resources.



Figure 14. Comparison of time taken for multiple GPUs to process 1600-million-pixel images in a single HPC node.

#### 4. CONCLUSION

We have developed a Mapping Application for Permafrost Land Environment (MAPLE) by combining Deep Learning, Big Imagery and HPC resources. Our workflow can run on heterogeneous HPC systems, demonstrating its interoperability for large scale implementation. We have tested the workflow with different HPC settings and compared the speed up. Three computational modules have been checked with the Frontera Longhorn HPC system.

The speed up achieved with multi-CPU-GPU model is 3.4x with 5 CPUs and 4 GPUs. When you increase the number of CPUs to 10 the speed increases to 4.0x only. This speed up means we can process an image with 33% of SUs with 5 images per batch according to the Frontera accounting model. The number of parallel processes can be employed depending on the amount of

179

main memory in the computing node. In Frontera-Longhorn that limit is 10 images per batch. But the size of satellite images can vary. It is safe to use 5 images per batch as the gain of the speedup is very small with 10 images per batch.

The multi-CPU-GPU model can be used with heterogeneous HPC systems effectively. But with a different HPC system the optimum number of images per batch can be different. We need to redo these computations to determine optimum number for each HPC system.

## ACKNOWLEDGEMENTS

This research was supported by the U.S. National Science Foundation grant #: 1720875, 1722572, 1927872, 1927723, 1927729. Supercomputing resources were provided by the eXtreme Science and Engineering Discovery Environment (Award #: DPP 190001) and Texas Advanced Computing Center Award #: DPP20001) Authors would like to thank Polar Geospatial Center, University of Minnesota for imagery support.

#### 5. REFERENCES

Amat, F., Höckendorf, B., Wan, Y. *et al.* Efficient processing and analysis of large-scale light-sheet microscopy data. *Nat Protoc* **10**, 1679–1696(2015).

Bhuiyan, M.A.E., Witharana, C. and Liljedahl, A.K., 2019. Big Imagery as a Resource to Understand Patterns, Dynamics, and Vulnerability of Arctic Polygonal Tundra. AGUFM, pp.C13E-1374.

El-Baz A, Suri JS. Big Data in Multimodal Medical Imaging. Boca Raton, FL: CRC Press; 2019

He, K.; Zhang, X.; Ren, S.2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, America, 27-30 Jun.

Jia Liu, Dustin Feld, Yong Xue, Jochen Garcke, Thomas Soddemann & Peiyuan Pan (2016) An efficient geosciences workflow on multi-core processors and GPUs: a case study for aerosol optical depth retrieval from MODIS satellite data, International Journal of Digital Earth, 9:8, 748-765

Kaiser, S., Grosse, G., Boike, J. and Langer, M. (2019): Lateral Thermokarst Lake Dynamics: A Remote Sensing Based Analysis of Spatial Variabilities and Erosion Processes, AGU Fall Meeting 2019, San Francisco, USA, 9 December 2019 - 13 December 2019.

Kanevskiy, M.; Shur, Y.; Jorgenson, T.; Brown, D.R.N.; Moskalenko, N.G.; Brown, J.; Walker, D.A.; Raynolds, M.K.; Buchhorn, M.2017. Degradation and stabilization of ice wedges: Implications for assessing risk of thermokarst in northern Alaska. Geomorphology, 297, 20–42.

Kremer J.; Stensbo-Smidt, K.; Gieseke, F.; S. Pedersen, K.; Igel, C.; "Big Universe, Big Data: Machine Learning and Image Analysis for Astronomy," in IEEE Intelligent Systems, vol. 32, no. 2, pp. 16-22, Mar.-Apr. 2017

Kucuk, A., Banda, J. & Angryk, R. A large-scale solar dynamics observatory image dataset for computer vision applications. Sci. Data 4, 170096 (2017).

Liljedahl, A.K.; Boike, J.; Daanen, R.P.; Fedorov, A.N.; Frost, G.V.; Grosse, G.; Hinzman, L.D.; Iijma, Y.; Jorgenson, J.C.; Matveyeva, N.; Necsoiu, M. 2016. Pan-Arctic ice-wedge degradation in warming permafrost and its influence on tundra hydrology. Nat. Geosci., 9, 312.

Muster, S.; Heim, B.; Abnizova, A.; Boike, J. 2013. Water body distributions across scales: a remote sensing based comparison of three arctic tundra wetlands. Remote Sens., 5, 1498–1523.

Nikolaou C. et al. (2014) Big, Linked and Open Data: Applications in the German Aerospace Center. In: Presutti V., Blomqvist E., Troncy R., Sack H., Papadakis I., Tordai A. (eds) The Semantic Web: ESWC 2014 Satellite Events. ESWC 2014. Lecture Notes in Computer Science, vol 8798. Springer, Cham.

Okunev, A.G.; Mashukov, M.Y.; Nartova, A.V.; Matveev, A.V. Nanoparticle Recognition on Scanning Probe Microscopy Images Using Computer Vision and Deep Learning. Nanomaterials 2020, 10, 1285.

Raynolds, M.K.; Walker, D.A.; Balser, A.; Bay, C.; Campbell, M.; Cherosov, M.M.; Daniëls, F.J.; Eidesen, P.B.; Ermokhina, K.A.; Frost, G.V.; Jedrzejek, B. A. 2019. raster version of the Circumpolar Arctic Vegetation Map (CAVM). Remote Sensing of Environment, 232,111297.2019.

Al-Saadi, A., Paraskevakos, I., Collares, B., Gonçalves, Heather J. Lynch, Shantenu Jha, Matteo Turilli Comparing 2020,Workflow Application Designs for High Resolution Satellite Image Analysis *arXiv:2010.14756* 

Schmied, C., Peter Steinbach, Tobias Pietzsch, Stephan Preibisch, Pavel Tomancak, An automated workflow for parallel processing of large multiview SPIM recordings, *Bioinformatics*, Volume 32, Issue 7, 1 April 2016, Pages 1112–1114

Skurikhin, A. N.; Wilson, C. J.; Liljedahl, A.; Rowland, J. C. 2014. Recursive active contours for hierarchical segmentation of wetlands in high-resolution satellite imagery of arctic landscapes. Southwest Symposium on Image Analysis and Interpretation, 137–140.

Steedman, A. E.; Lantz, T. C.; Kokelj, S. V.2017. Spatiotemporal variation in high-centre polygons and ice-wedge melt ponds, Tuktoyaktuk coastlands, Northwest Territories. Permafr. Periglac. Process., 28, 66–78.

Zhang, W.; Witharana, C.; Liljedahl, A.K.; Kanevskiy, M. 2018. Deep Convolutional Neural Networks for Automated Characterization of Arctic Ice-Wedge Polygons in Very High Spatial Resolution Aerial Imagery. Remote Sens, 10, 1487.