An Optimal GeoAI Workflow for Pan-Arctic Permafrost Feature Detection from High-Resolution Satellite Imagery

Mahendra R. Udawalpola, Amit Hasan, Anna Liljedahl, Aiman Soliman, Jeffrey Terstriep, and Chandi Witharana

Abstract

*High-spatial-resolution 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. Knowledge discovery through artificial intelligence, big imagery, and high-performance computing (*HPC*) resources is just starting to be realized in Arctic permafrost science. 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 permafrost modeling. Interoperability across heterogeneous* HPC *systems and optimal usage of computational resources are key design goals of* MAPLE*. We systematically compared the performances of four different* MAPLE *workflow designs on two* HPC *systems. Our experimental results on resource utilization, total time to completion, and overhead of the candidate designs suggest that the design of an optimal workflow largely depends on the* HPC *system architecture and underlying service-unit accounting model.*

Introduction

Big image-data analysis has become essential in an array of scientific applications, such as computer vision (Kucuk *et al.* 2017), medical imaging (El-Baz and Suri 2020), materials science (Okunev *et al.* 2020), and astronomy (Kremer *et al.* 2017). The advancements of satellite sensor technology, coupled with the ever-increasing spatial resolution and temporal frequency of image acquisitions, ideally position remote sensing applications in the big-data landscape (Wang *et al.* 2015; P. Liu *et al.* 2018). Satellite imagery archives are being radically transformed from terabytes to petabyte scale (Witharana *et al.* 2021). The sheer volumes of imagery pose new challenges in storage, analysis, and visualization techniques (P. Liu 2015; Y. Ma *et al.* 2015), and the requirements exceed the capabilities of existing general-purpose computing resources. Therefore, highly efficient workflows with high-performance computing resources are required for implementing big-imagery applications.

High-throughput computing (HTC) and high-performance computing (HPC) are both important in high-resolution imagery analysis on a petabyte scale. HTC is used for workloads that consist of tasks that are independent of each other and can start or complete in any order (e.g., automated feature extraction from thousands of satellite images in repeated mapping applications). Therefore, there is a lot of flexibility in scheduling these HTC jobs in HPC systems. In contrast, an HPC workload is characterized by its scalability or running time. Typically, an HPC workload consists of a single job that coordinates multiple processes which run at the same time. When running these jobs, input–output requirements are important. Usually, HTC tasks operate on a small volume of data and HPC workloads operate on large volumes of data. But in running many HTC jobs, the limitations of input–output bandwidth become significant. Usually, most supercomputers are designed for HPC workloads. Huerta *et al.* (2019) argue that new applications require a paradigm shift in computing architecture to address large data sets, deep-learning algorithms, and hybrid workloads using both HPC and HTC. It is imperative to find out how applications with hybrid workloads can be run efficiently in existing HPC resources. Remote sensing (RS) big-data applications typically consist of hybrid workloads requiring efficient use of existing HPC systems. Lee *et al.* (2011) reviewed advances in HPC applied to remote sensing problems, and in particular HPC-based platforms, such as multi-processor systems and large-scale and heterogeneous networks of computers.

A seamless application of HPC resources for translating big satellite imagery into science-ready products can enable knowledge discovery at the nexus of the human and natural systems (Chi *et al.* 2016). In recent years, the use of HPC resources has become an inextricable component in big-imagery applications (Wang *et al.* 2018). A plethora of applications can be found in the literature involving big imagery and HPC. Amat *et al.* (2015) developed a workflow for light-sheet microscopy, which involves several tens of terabytes of data. Schmied *et al.* (2016) compared the performance of an automated workflow on a single workstation and an HPC cluster. J. Liu *et al.* (2016) analyzed a geosciences workflow on multi-core processors and graphical processing units (GPUs), achieving a 5× speedup on a multi-core processor and a 43× speedup for some parts of the workflow on GPU. In a recent study, Al-Saadi *et al.* (2021) compared workflow application designs for high-resolution satellite-imagery analysis. They analyzed three workflow designs using the Extreme Science and Engineering Discovery Environment (XSEDE) HPC system for two use cases, for a total of 4672 high-resolution satellite and aerial images and 8.35 TB of data.

Modern HPC systems consist of many HPC computer nodes. Each node contains multi-core central processing units (CPUs) and multi-GPUs. RS big-data applications need to use both CPUs and GPUs in their workflow, because GPUs are efficient at processing RS images and CPUs are efficient at executing complex algorithms. Several traditional parallel paradigms are widely used in these systems, such as OpenMP and Message Passing Interface. Implementation of parallel RS algorithms using Message Passing Interface is difficult, and HPC systems are not optimized for data-intensive computing (Wang et. al. 2016). RS workloads involve both HPC and HTC features, so they are considered hybrid HPC/HTC workloads. A single RS workload may not be large enough for use in many multiple nodes. It is therefore critical to examine how to optimize RS hybrid HPC/HTC workloads in a single node with

---

Mahendra R. Udawalpola, Amit Hasan, and Chandi Witharana are with the Department of Natural Resources and the Environment, University of Connecticut (mahendra.udawalpola@uconn.edu).

Anna Liljedahl is with the Woodwell Climate Research Center, Falmouth, MA.

Aiman Soliman and Jeffrey Terstriep are with the National Center for Supercomputing Applications, University of Illinois Urbana–Champaign.

Contributed by Alper Yilmaz, August 30, 2021 (sent for review September 17, 2021).

Photogrammetric Engineering & Remote Sensing
Vol. 88, No. 3, March 2022, pp. 1–xxx.
0099-1112/22/1–xxx

doi: 10.14358/PERS.21-00059R2

multi-CPU and multi-GPU cores. The use of HPC resources is measured using service units (SUs). RS workflows with big-imagery analysis need to be optimized for both Sus and running time. Different HPC systems measure the use of SUs by different accounting models. These different configurations present new challenges in designing efficient workflows for targeted applications which require both CPU and GPU processing.

Traditional remote sensing image-analysis algorithms fail to grapple with the image complexities and high-level semantics arising from sub-meter-resolution satellite imagery (Blaschke 2010; Blaschke *et al.* 2014; Lang *et al.* 2018). 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 (L. Ma *et al.* 2019). Deep-learning convolutional neural nets (DLCNNs; LeCun *et al.* 2015) have shown great potential for semantic object-instance segmentation in detecting and delineating each distinct object in an image of common objects from everyday images. The success of DLCNNs in computer-vision applications has received great interest from the remote sensing community (L. Ma *et al.* 2019). But DLCNN algorithms are computationally intensive and demanding of memory. Thus, it is important to optimize data management, image processing, classification, and visualization techniques, because they serve as key bottlenecks in image-to-assessment pipelines.

Archived observation data was predicted by the Open Geospatial Consortium to exceed an exabyte by 2015 (Karmas *et al.* 2016). But it is estimated that up to 95% of the data present in existing archives have never been accessed (Nikolaou *et al.* 2014). Over the last decade, the entire Arctic has been imaged at 0.5-m resolution several times by commercial satellite sensors of Maxar Technologies (previously known as DigitalGlobe; Witharana *et al.* 2020). The image repository at the Polar Geospatial Center at the University of Minnesota provides transformational opportunities to observe, monitor, and document permafrost thaw occurring across the Arctic tundra, which is a logistically challenging region with an extremely sparse field observation network. Landscapes of Alaska, Canada, and Russia harbor approximately 5 million km2 of tundra. But imagery is underutilized, and derived science products are rare despite their unprecedented potential for pan-Arctic permafrost monitoring and modeling applications.

Permafrost—unique landscapes comprising Earth materials that remain at or below 0°C for at least two consecutive years—covers approximately 24% of the exposed land surface of the Northern Hemisphere (Brown *et al.* 1997). Ice-rich permafrost can be identified by atypical surface features called ice-wedge polygons (IWPs), which are underlain by ice wedges several meters wide and deep that form a connected network across the tundra (Kanevsky *et al.* 2016). Vegetation and geology maps suggest that about two-thirds or more of the Arctic landscape is occupied by polygonal ground (Kokelj *et al.* 2015; Raynolds *et al.* 2019) and therefore ice-rich ground, but the exact extent and the prevailing IWP types (i.e., whether the ice wedges experience melt or not) are largely unknown.

Over recent decades, ice-wedge degradation—the transformation of low-centered polygons into high-centered polygons—has been documented at several locations across the Arctic tundra in the field and through localized remote sensing analyses (Liljedahl *et al.* 2016; 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 unusually warm summers, wildfires, or human activities initiating the onset of ice-wedge degradation (Jorgenson *et al.* 2006; Jones *et al.* 2015; Raynolds *et al.* 2020). Degradation of ice wedges is a quasi-cyclic process, often occurring over a shorter time scale than the formation of new permafrost (aggradation), with the latter controlled by the accumulation of organic and mineral soil above the ice wedge (Kanevskiy *et al.* 2017). Understanding the spatiotemporal dynamics behind the evolution of ice-wedge polygonal tundra demands objective and detailed maps consolidating the extent of ice wedges and their prevailing successional stages (Witharana *et al.* 2021).

Despite the alarming signals, the Arctic science community has a limited understanding of the spatiotemporal continuity of these otherwise locally observed changes. The lack of knowledge about the larger geographical extent and successional stage of IWPs introduces 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 in varying temporal windows (Nitze *et al.* 2018; Witharana *et al.* 2019). IWPs are difficult to detect in any remote sensing imagery with a spatial resolution coarser than 4 m (Muster *et al.* 2012). Sub-meter-resolution commercial satellite imagery (e.g., Maxar) demonstrates greater promise for 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 for delineating polygons from high-resolution remote sensing imagery (Witharana *et al.* 2021). A considerable number of local-scale studies have analyzed ice-wedge degradation processes using satellite imagery, as well as imagery and lidar data from manned and unmanned aerial vehicles (Muster *et al.* 2013). Most studies to date have relied on manual image interpretation or semi-automated approaches (Skurikhin *et al.* 2014) and been confined to site-to-local-scale mapping. Therefore, there is a need and an opportunity to use very-high-spatial-resolution imagery in regional-scale mapping efforts to spatiotemporally document microtopographic changes due to thawing ice-rich permafrost.

Despite the remarkable performance of DLCNNs in everyday image understanding, bottlenecks still exist in the translation to geo-object detection from remote sensing imagery. Image dimensions, multiple spectral channels (more than the standard red, green, and blue channels), spatial reference, seasonality, and most importantly the semantic complexity of geo-objects aggregated into multiple spatial scales impose greater friction on the inferential strength of DLCNN model predictions. The scalability of automated analysis over millions of square kilometers comprising heterogeneous landscapes reinforces the need for efficient workflows. To surmount these challenges, we have developed a novel image-to-assessment pipeline—Mapping Application for Arctic Permafrost Land Environment (MAPLE)—which can be deployed in heterogeneous supercomputing resources. MAPLE is a first-of-its-kind pan-Arctic mapping effort that attempts to characterize microtopography using sub-meter-resolution imagery without compromising geographical extent.

The overarching goal of MAPLE is to produce the first pan-Arctic IWP map using a large volume of commercial satellite imagery available at the Polar Geospatial Center and HPC resources from computing facilities funded by the US National Science Foundation. In the first stage, we will produce a circumpolar IWP map for the regions that have been identified as high-probability ground ice content by Brown *et al.* (2002). Then we will progressively extend the mapping to medium- and low-probability ground ice areas of Brown *et al.*, ultimately covering the entire tundra. The ongoing mapping area (Figure 1) includes around 25 000 satellite images and over 180 TB of data.

The main objective of this article is to analyze the computational efficiency of the MAPLE workflow in heterogeneous HPC environments, which involve both CPUs and GPUs. We further aim to understand how different workflow designs interact with underlying SU accounting models of the HPC systems, which in turn support optimal resource usage to complete image-analysis problems at hand.

Methods

Mapping Application for Arctic Permafrost Land Environment (MAPLE)

Figure 2 shows a generalized framework for high-performance image analysis with MAPLE using imagery from the Polar Geospatial Center and computing resources from multiple computing environments, such as Frontera at the Texas Advanced Computing Center and XSEDE, to produce science-ready products. The MAPLE workflow (Figure 3) is threefold: image preprocessing, DLCNN prediction (inferencing), and postprocessing. While the first and last segments involve CPU implementations, prediction can operate on GPUs or CPUs. MAPLE takes

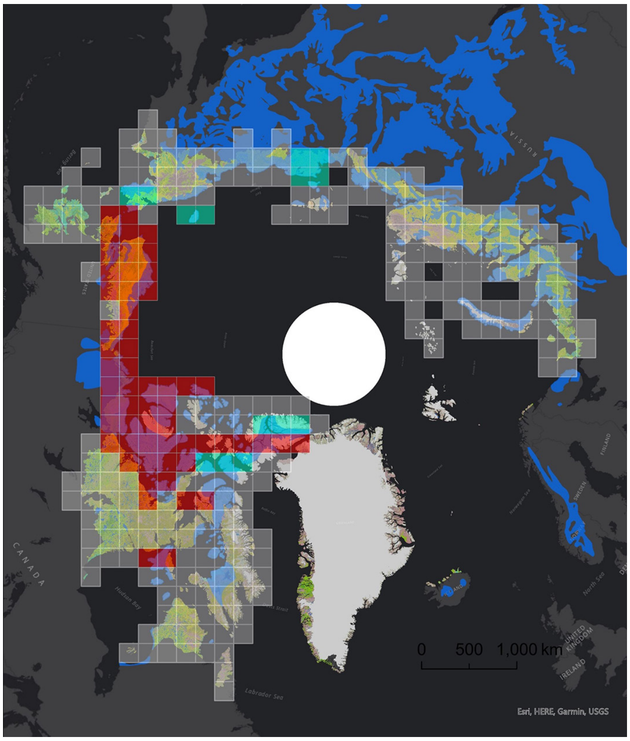

Figure 1. Ongoing deployment of Mapping Application for Arctic Permafrost Land Environment (MAPLE) in Arctic polygonal tundra. The map is overlain by the circumpolar Arctic vegetation map of Raynolds *et al.* (2019) and the high-probability ground ice map of Brown *et al.* (2002). Colored grid cells (200×200 km) represent the progress of the mapping. Red and dark-green squares represent completed areas and areas in progress, respectively. Gray squares represent the area to be mapped. Blue shading represents the Circum-Arctic Map of Permafrost and Ground-Ice Conditions, Version 2 (Brown *et al.* 2002), and light-green shading represents the Circumpolar Arctic Vegetation Map (Raynolds *et al.* 2020), which were used as guides to prioritize mapping areas.

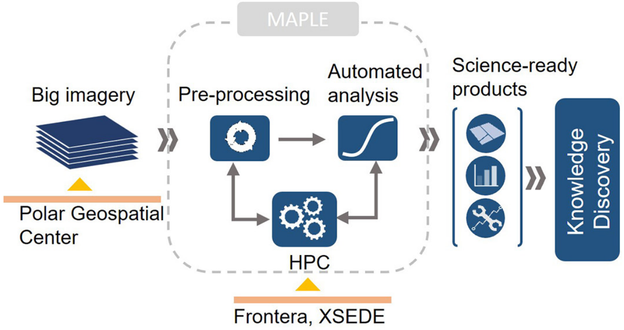

Figure 2. General semantic diagram for high-performance analysis of very-high-spatial-resolution satellite imagery using Mapping Application for Arctic Permafrost Land Environment (MAPLE). The images are obtained from the Polar Geospatial Center at the University of Minnesota, then processed by the MAPLE workflow using high-performance computing (HPC) resources from the Frontera system at the Texas Advance Computing Center and the Bridges system from the Extreme Science and Engineering Discovery Environment (XSEDE). The ice-wedge polygon map and the surface water-body map serve as the two key science-ready products of the workflow.

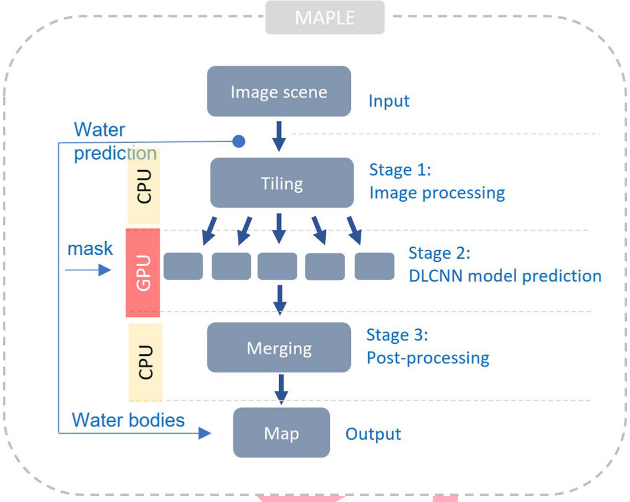

Figure 3. General semantic diagram of the Mapping Application for Arctic Permafrost Land Environment (MAPLE) workflow. MAPLE is a modular workflow consisting of three stages relying on both central processing unit (CPU) and graphical processing unit (GPU) resources. Stage 1 is a CPU-based implementation, which involves two operations: automated extraction of surface water bodies and tiling of the input image scene into small patches for use in later stages. Stage 2 is a GPU-based operation. It implements the deep-learning convolutional neural net (DLCNN) algorithm for predicting ice-wedge polygons. In stage 3, stitching of shape files and removal of duplicates are performed using CPU resources. The final output of the MAPLE workflow are the ice-wedge polygon map and surface water-body map of the input image scene.

high-resolution satellite images as input and outputs two key geospatial layers: ice-wedge polygons and surface water bodies. The spatial resolution of satellite imagery is acquired at 0.5 m, comprising multispectral channels (blue, green, red, near-infrared, or more depending on the sensor) at 16-bit radiometric resolution with a typical footprint of 20×20 km (i.e., 160 million pixels/image). At the initial stage, we process the high-probability region of the IWP map using MAPLE. Figure 1 shows the progress of MAPLE deployment in polygonal tundra.

We use Mask RCNN (He *et al.* 2016) as the key DLCNN model in MAPLE. DLCNN models show better performance in GPUs than in CPUs. Usually, the amount of memory available in GPUs (on average, 16 GB) is much smaller than that in CPUs. Therefore, we cannot perform DLCNN operations for the complete satellite image in GPU memory. Due to this limitation, we need to split the satellite image, which is around 5 GB on disk (40 000×40 000 pixels) into small tiles (200×200 pixels). To alleviate any terminological ambiguities, throughout this article we will use the term *image scene* to refer to an entire satellite image and the term image tile to refer to a subsetted array obtained by tiling the *image scene* according to predefined tile dimensions. Due to this splitting, the ice-wedge polygons can be intersected by the boundaries and may be missed in the prediction stage. To alleviate this effect, we keep a 10% overlap between image tiles, resulting in more than 40 000 image tiles per image scene. These image tiles are saved using a compressed HDF5 binary data format, after excluding water bodies and no-data areas. Subsequently, compressed tiles are accessed in parallel within the workflow and output detected ice-wedge polygons. Each parallel process stores predicted ice-wedge polygons as an individual shape file.

In the postprocessing stage (stage 3), the shape files generated by each GPU are stitched into a single shape file. The resulting shape file contains duplicate polygons along the seams of images tiles due to the 10% overlap. We remove those duplicate polygons during this stage. Figure 4 depicts automated mapping results of ice-wedge polygons and water bodies from example locations comprising different tundra types in Alaska and Canada.

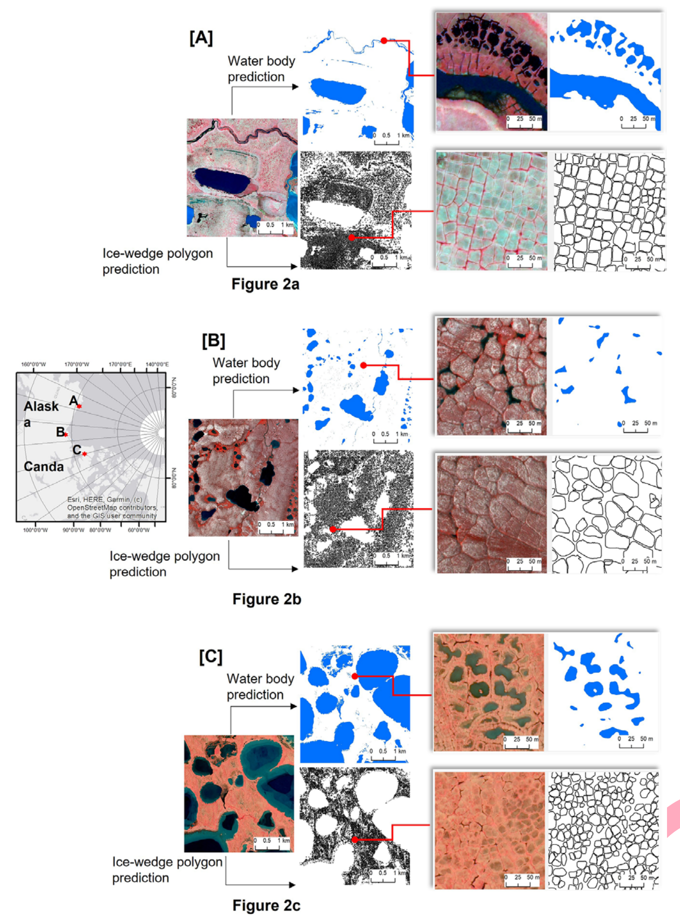

Figure 4. Sample input image and output ice-wedge polygons and water bodies of different scales from the Mapping Application for Arctic Permafrost Land Environment (MAPLE) workflow for (a) the North Slope of Alaska, and (b and c) Canada (see inset map at left). In each subfigure, the leftmost image shows the high-resolution satellite image. The second column shows the water-body map (blue) and ice-wedge polygon map of the corresponding area. The third column shows a zoomed view of two separate areas of the previous image patch. The rightmost column shows the water-body map (blue) and ice-wedge polygon map of the previous images. Imagery © 2016 DigitalGlobe, Inc.

An efficient workflow is required, because we need to process thousands of high-resolution satellite images. We use several techniques to optimize our workflow. The first is to minimize the area of processing by removing large numbers of water bodies. The Arctic region contains many water bodies, on scales from sub-meter to hundreds of meters. The MAPLE workflow first detects these water bodies using techniques developed by Kaiser *et al.* (2021). The predicted water from stage 1 of the workflow is used as a precursor layer for tiling the image. This will avoid unnecessary implementation of the IWP prediction algorithm on water areas in stage 2. Iterative prediction and use of the water mask not only produces a sub-meter-scale map of surface water but also speeds up the IWP prediction. The second optimization strategy is to remove image overlaps. A given satellite footprint has a considerable number of spatial overlaps with its neighbors, due to different imaging times and different sensors. We can significantly reduce the processing volume and computing resources (caused by duplicate application of the DLCNN model on the same spatial locales) by removing these image overlaps. We have developed an algorithm that calculates the image overlaps and excludes them from processing in our workflow for a given image footprint.

Model Training

We used a transfer learning strategy to retrain the Mask RCNN network. Using the online tool VGG Image Annotator, an annotated ice-wedge polygon data set was created 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 pretrained weights generated by the COCO data set and trained only the head layers of the Mask RCNN network. The training was implemented using an NVIDIA GeForce RTX 2080 GPU with 10 GB of memory. We trained the Mask RCNN model with a mini-batch size of two image tiles, 250 steps per epoch, a learning rate of 0.001, a learning momentum of 0.9, a weight decay of 0.0001, and 50 epochs.

Workflow Designs

Modern HPC resources such as Frontera and XSEDE consist of multiple nodes. Each node contains multiple CPUs and GPUs. Each CPU and GPU contains multiple cores. Programs should be designed to use these resources optimally. Figure 5 shows a semantic diagram of the sequential workflow (design 1) in a single computing node. In this setup, we do not use multiple CPUs and GPUs available in the node. The three stages of preprocessing, inferencing, and postprocessing are executed sequentially.

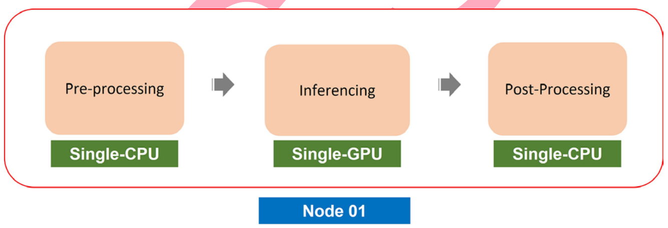

Figure 5. A semantic diagram of a single-CPU (central processing unit), single-GPU (graphical processing unit) Mapping Application for Arctic Permafrost Land Environment (MAPLE) workflow design using a single high-performance computing node (design 1). The workflow analyzes a single image at a time in three stages. The first stage (preprocessing) and the last (postprocessing) are executed on a CPU, and the second stage is executed on a GPU.

Figure 6 shows a semantic diagram of the design in which multiple GPUs in a single computing node are used in the inferencing stage. Here the image tiles generated in the preprocessing stage are stored in a single multi-threaded queue and processed using multiple GPUs on a single node.

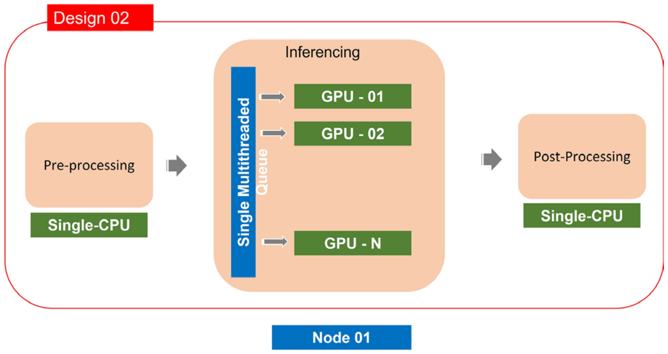

Figure 6. A semantic diagram of a single-CPU (central processing unit), multi-GPU (graphical processing unit) Mapping Application for Arctic Permafrost Land Environment (MAPLE) workflow design using a single high-performance computing node with a single multi-threaded queue for all GPU cores (design 2). A single image is preprocessed using a single multi-core CPU and sent to multiple GPUs for detection using a single multi-threaded image queue. The detected ice-wedge polygons are postprocessed using a single multi-core CPU.

Figure 7 shows a semantic diagram of a design similar to the one in Figure 6. In this workflow, we use a dedicated queue for each GPU core, as illustrated. The image tiles generated in the preprocessing stage are distributed among separate queues and then inferenced by a dedicated GPU core. Figure 8 shows a multi-CPU, multi-GPU workflow design in which we processed multiple images per batch (design 4). In preprocessing, we use multiple CPU cores in a single node. Then tiles from each image are processed in separate nodes for the inferencing stage using the optimum number of GPUs available in that node. The shape files generated in the inferencing stage are processed in a single HPC node using multiple CPU cores in the postprocessing stage.

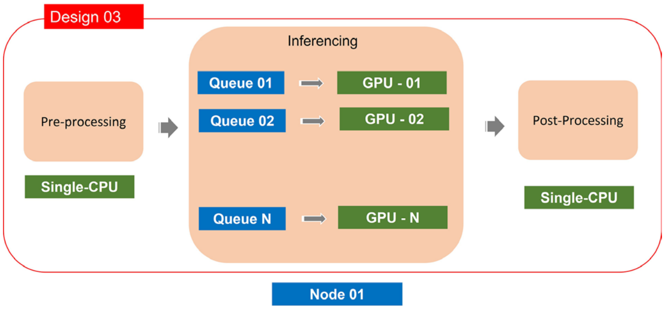

Figure 7. A semantic diagram of a single-CPU (central processing unit), multi-GPU (graphical processing unit) Mapping Application for Arctic Permafrost Land Environment (MAPLE) workflow design using a single high-performance computing node with a dedicated queue for each GPU core (design 3). A single image is preprocessed using a single multi-core CPU and sent to multiple GPUs for detection of ice-wedge polygons using a dedicated image queue for each GPU. The detected ice-wedge polygons are postprocessed using a single multi-core CPU.

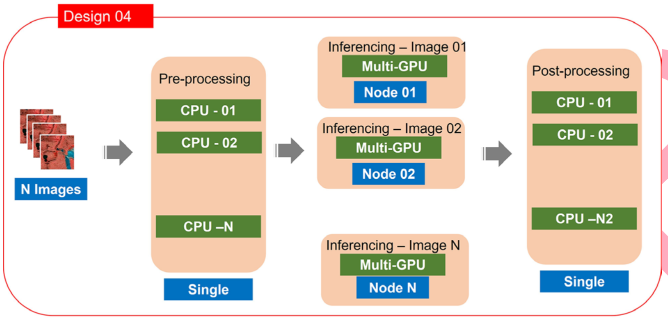

Figure 8. A semantic diagram of a multi-CPU (central processing unit), multi-GPU (graphical processing unit) Mapping Application for Arctic Permafrost Land Environment (MAPLE) workflow design using multiple high-performance computing nodes with a dedicated queue for each GPU core (design 4). A set of images is preprocessed using multiple CPU cores in a single node and sent to multiple GPUs in different nodes for detection of ice-wedge polygons, with a dedicated image queue for each GPU. The detected ice-wedge polygons are postprocessed using a single node with multiple CPU cores.

Numerical Experiments

The numerical experiments were carried out on the Frontera Longhorn HPC system computing nodes and XSEDE Bridges2 computing nodes. The former consists of 96 computing nodes and the latter of 24 computing nodes. Node specifications are listed in Table 1.

The effective use of HPC resources depends on the underlying resource accounting model, HPC architecture, and workflow design. Project resources are allocated based on SUs. In the Frontera Longhorn system, one SU is calculated by multiplying the job duration in wall-clock hours, the charge rate per node hour, and the number of nodes per job. Therefore, to get maximum resource use we need to use all four GPUs per job. In contrast, on XSEDE Bridges2, one SU is calculated by multiplying the job duration, the number of GPUs per node, the charge rate per hour, and the number of nodes. Here we must calculate the optimum number of GPUs for a single job. We can optimize our workflow based on node time and SUs. In Frontera Longhorn, these two are proportional, but in XSEDE Bridges2, the optimum workflows for time and SUs can be different. We will examine the four different MAPLE workflow designs illustrated in Figures 5 through 8 to find the optimum designs on different HPC systems.

Table 1. Computing node configurations for Frontera Longhorn (Texas Advanced Computing Center) and Extreme Science and Engineering Discovery Environment (XSEDE) Bridges2.

System	Frontera Longhorn	XSEDE Bridges2
Processor	IBM Power 9	Intel Xeon Gold 6248
Total Processors/Node	2	2
Total Cores/Processor	20	20
Total Cores/Node	40	40
Clock Rate (GHz)	2.3	2.5
RAM (GB)	256	512
GPUs/Node	4 × NVIDIA Tesla V100	8 × NVIDIA Tesla V100
GPU RAM/Core (GB)	16	32

GPU = graphical processing unit; RAM = random-access memory.

Results and Discussion

We evaluated the time taken for three stages of the sequential workflow (design 1) described in Figure 5 for different image dimensions as a base case. Figure 9 shows the computation results for Frontera: preprocessing time, inferencing time, postprocessing time, and total time for images with different sizes on a CPU or a GPU. The gray bars show the time taken for a 400-million-pixel image on a CPU. Orange, green, and brown bars respectively, represent 400-, 1600-, and 3600-million-pixel images on a GPU. The computation time depends on the image size as well as the number of polygons detected. Comparing the first two bars for inferencing, it is evident that using a GPU for inferencing achieves a 9.0× speedup. The reason for this speedup is that DLCNN computations can be performed in parallel with many GPU cores. Increasing the size of the image increases the time in all stages. The time taken to process a 3600-million-pixel image on a GPU is on the same order as the time taken to process 400 million pixels using only a CPU.

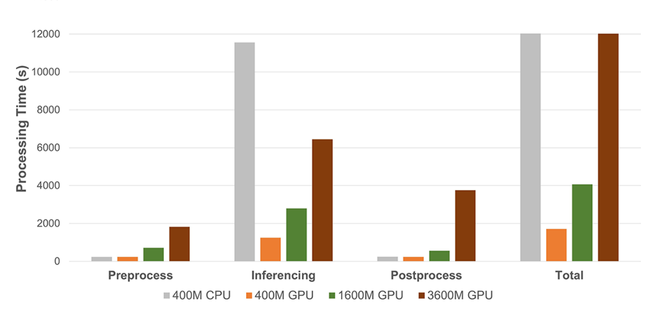

Figure 9. Comparison of time taken for Mapping Application for Arctic Permafrost Land Environment (MAPLE) workflow design 1 on Frontera for different image sizes on CPUs and GPUs. Gray represents the computation times for a 400-million-pixel image on a CPU. Orange, green, and brown represent the computation times for a 400-million-, 1600-million-, and 3600-million-pixel image, respectively, on a GPU. The first three groups of bars show computation times for preprocessing, inferencing, and postprocessing stages, respectively. The last group shows the total time taken to process each image.

Figures 10 (Frontera Longhorn) and 11 (XSEDE Bridges2) show the comparison of inferencing times for designs 2 and 3 with a 160-million-pixel image. The use of a dedicated queue in design 3 improves the running time for all four cases. Using four GPUs, we manage to obtain a 3.6× speed up in Frontera and a 2.0× speedup in XSEDE. The perfect speedup cannot be obtained because of input–output operations and serial sections in the workflow.

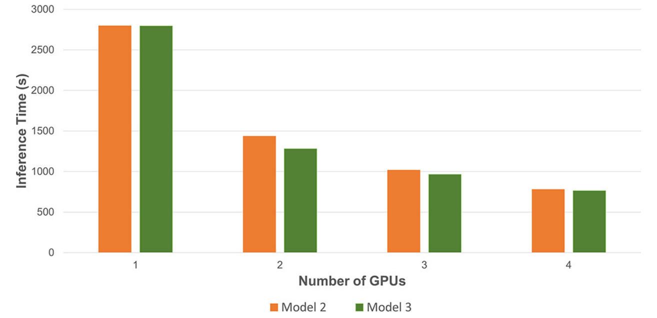

Figure 10. Comparison of inferencing times for multiple GPUs in a single Frontera Longhorn HPC node to process a 1600-million-pixel image with design 2 (orange) and design 3 (green). GPU = graphical processing unit; HPC = high-performance computing.

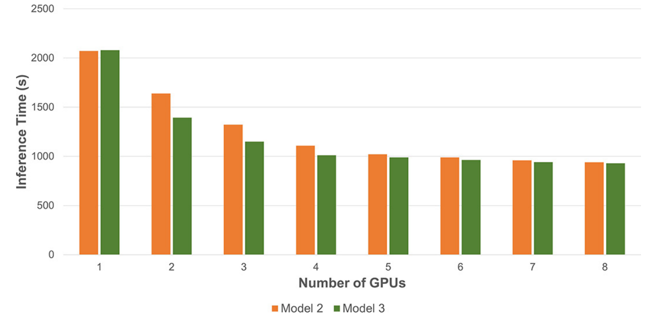

Figure 11. Comparison of inferencing times for multiple GPUs in a single XSEDE Bridges2 HPC node to process a 1600-million-pixel image with design 2 (orange) and design 3 (green). GPU = graphical processing unit; HPC = high-performance computing.

Service units are important in processing large numbers of images, because the number of computing resources is limited. In Frontera, running time and service units consumed are proportional because of its accounting model; therefore we will only present results for running time in Frontera. But in XSEDE those two can be different, so we examine both service units and running time in that system.

Figure 12 shows the service units consumed by inferencing a 1600-million-pixel image with up to eight GPUs in XSEDE with designs 2 and 3. The most efficient use of service units can be obtained with a single GPU—that is, design 1. But with the four GPUs, we can achieve a 2× speedup with 2.2 service units.

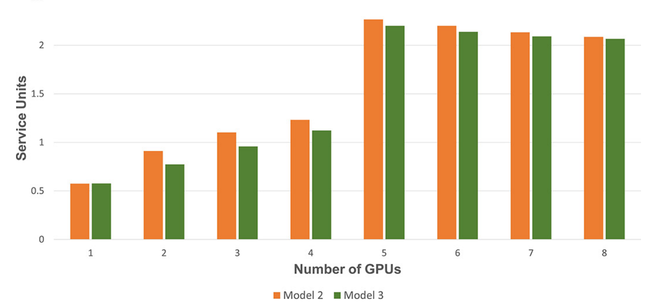

Figure 12. Comparison of service units used in inferencing a 1600-million-pixel image with multiple GPUs in a single XSEDE Bridges2 HPC node with design 2 (orange) and design 3 (green). GPU = graphical processing unit; HPC = high-performance computing.

Figure 13 shows the computation times for a full workflow (design 3) for a 1600-million-pixel image with up to four GPUs in Frontera. The speedup achieved for a full workflow using four GPUs is 2.0×. The reason for the lower speedup is the increase in the percentage of serial workload. Figure 14 shows the running times for design 3 in XSEDE, where we can use up to eight GPUs per node. The speedup achieved with multiple GPUs saturates at 1.5× with four GPUs.

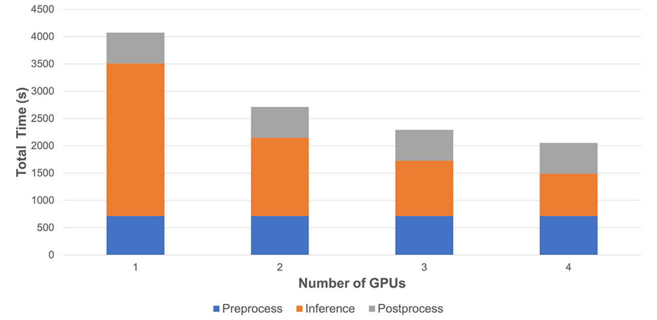

Figure 13. Comparison of times taken for multiple GPUs to process a 1600-million-pixel image in a single Frontera HPC node with design 3. Preprocessing time, inferencing time, and postprocessing time are represented by blue, orange, and gray, respectively. GPU = graphical processing unit; HPC = high-performance computing.

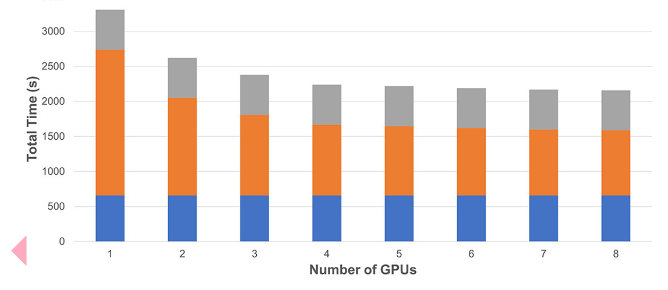

Figure 14. Comparison of times taken for multiple GPUs to process a 1600-million-pixel image in a single XSEDE HPC node with design 3. Preprocessing time, inferencing time, and postprocessing time are represented by blue, orange, and gray, respectively. GPU = graphical processing unit; HPC = high-performance computing.

Figure 15 shows the total SUs consumed in XSEDE when using up to eight GPU cores. The most efficient use of SUs occurs when the code is run with one GPU core. But we can get a 1.5× speedup for four GPUs with 1.75× SU use. Figure 16 shows the total amount of time taken to process one 1600-million-pixel image using design 4 with four GPUs in a Frontera HPC node. Preprocessing and postprocessing are done using multiple CPU cores in a GPU node. The speedup using 10 CPU cores in a single computing node for preprocessing is 7.5×, and for postprocessing it is 9.7×, which results in a combined speedup of 8.4×. Preprocessing is a memory-intensive task. It needs four times the memory of the image. With 256 GB RAM available in one node, we can only process up to 10 1600-million-pixel images. The first bar of Figure 16 shows the result we obtain with design 3. A speedup of 2.4× with five CPUs and 2.9× is achieved with design 4 compared to design 3 with a full workflow. Compared with design 1, a speedup of 3.4× with five CPUs and 4.0× with 10 CPUs is achieved. This 4.0× speedup means we can process four times faster than design 1 with the same resources in Frontera.

Figure 17 shows the total time taken with design 4 in XSEDE. Preprocessing and postprocessing are done using up to 10 CPUs cores per image batch and four GPUs per image. The speedup saturates after four CPU cores. Figure 18 shows the SUs consumed to process images with design 4 using up to 10 CPU cores per image batch and a single GPU core per image. The SUs at first decrease slightly with the increase of CPU cores, but increase again after four cores due to the restriction that only four CPUs are allowed for allocation to one unit (GPU) in the shared GPU queue. If we want to allocate more than four CPU cores, we need to allocate two GPUs. This increases the SUs used by the calculations, because SUs are proportional to the number of GPUs. The same happens when we use more than eight CPU cores.

ConclusionWe developed the 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 tested the workflow with different HPC settings and compared the

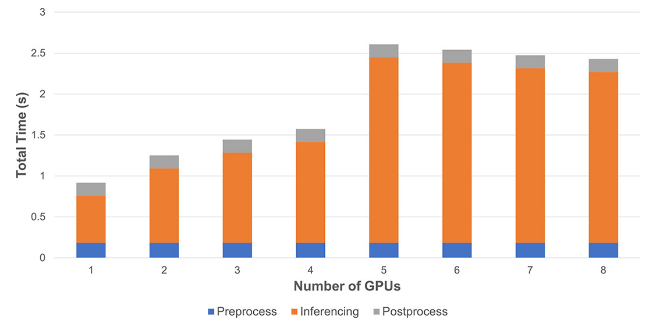

Figure 15. Comparison of service units used for multiple GPUs to process a 1600-million-pixel image in a single XSEDE HPC node with design 3. Preprocessing time, inferencing time, and postprocessing time are represented by blue, orange, and gray, respectively.

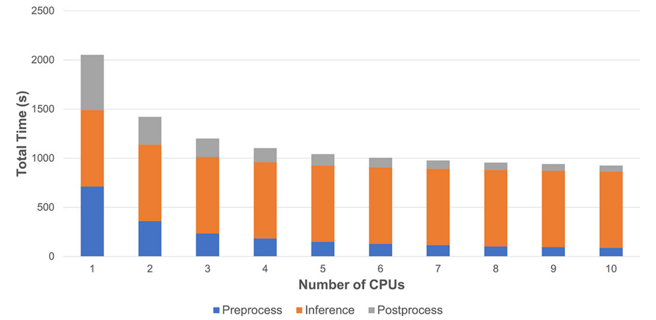

Figure 16. Comparison of times taken to process 1600-million-pixel images in the Frontera HPC system using design 4 with four GPUs per image. A batch of images is sent for preprocessing using a multi-core CPU and then inferencing is done using four GPUs per image. Then postprocessing is performed using a multi-core CPU. Preprocessing time, inferencing time, and postprocessing time are represented by blue, orange, and gray, respectively. CPU = central processing unit; GPU = graphical processing unit; HPC = high-performance computing.

speedup and resource utilization. Four workflow designs were checked with the Frontera Longhorn and XSEDE Bridges2 HPC systems.

The speedup achieved with design 4 in Frontera is 3.4× with five CPUs and four GPUs. The number of parallel processes that can be used depends on the amount of main memory in the computing node. The pipeline saturates in Frontera after four GPUs and five CPUs. It is safe and effective to use five images per batch, as the gain in speedup is very small with 10 images (which is the memory limit) per batch. We can process an image with 33% of SUs with five images per batch in design 4 compared to design 1, according to the Frontera accounting design. The speedup achieved in XSEDE design 4 is 2× for the full workflow, which is obtained with five CPUs and four GPUs. But this will use more SUs per single image, due to the XSEDE accounting model. In XSEDE, design 4 uses the fewest SUs with four images per batch and one GPU per image.

The multi-CPU, multi-GPU design can be used effectively with heterogeneous HPC systems. Design 4 is the fastest of all the designs. But with a different HPC system, the optimum number of images per batch (CPUs) can be different. Design 4 also has the lowest SU usage with different numbers of GPUs per image. The resource usage can be different with different HPC systems due to the accounting design and system architecture. Therefore, we need to test design 4 in different HPC systems to find out optimum CPU and GPU combinations before doing large-scale calculations to optimize our workflow. Design 4 is suitable for big-imagery GeoAI workflows such as MAPLE in existing HPC systems.

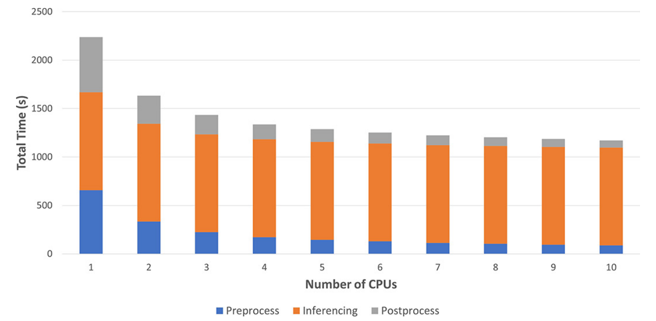

Figure 17. Comparison of times taken to process 1600-million-pixel images in the XSEDE HPC system with different number of CPUs per image batch and four GPUs per image, using design 4. A batch of images is sent for preprocessing using a multi-core CPU, and then inferencing is done using four GPUs per image. Then postprocessing is performed using a multi-core CPU. Preprocessing time, inferencing time, and postprocessing time are represented by blue, orange, and gray, respectively. CPU = central processing unit; GPU = graphical processing unit; HPC = high-performance computing.

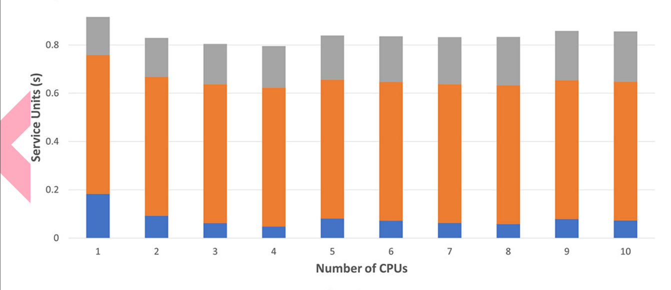

Figure 18. Comparison of service units used to process 1600-million-pixel images in the XSEDE HPC system with different numbers of CPUs per image batch and a single GPU per image, using design 4. A batch of images is sent for preprocessing using a multi-core CPU, and then inferencing is done using one GPU per image. Then postprocessing is performed using a multi-core CPU. Preprocessing time, inferencing time, and postprocessing time are represented by blue, orange, and gray, respectively. CPU = central processing unit; GPU = graphical processing unit; HPC = high-performance computing.

Acknowledgments

This research was supported by the US National Science Foundation grants 1720875, 1722572, 1927872, 1927723, and 1927729. Supercomputing resources were provided by the Extreme Science and Engineering Discovery Environment (award DPP 190001) and Texas Advanced Computing Center (award DPP20001). The authors would like to thank the Polar Geospatial Center at the University of Minnesota for imagery support.

References

Al-Saadi, A., I. Paraskevakos, B. C. Gonçalves, H. G. Lynch, S. Jha and M. Turilli. 2021. Comparing workflow application designs for high resolution satellite image analysis. *Future Generation Computer Systems* 124:315–329.

Amat, F., B. Höckendorf, Y. Wan, W. C. Lemon, K. McDole and P. J. Keller. 2015. Efficient processing and analysis of large-scale light-sheet microscopy data. *Nature Protocols* 10:1679–1696.

Bhuiyan, M.A.E., C. Witharana and A. K. Liljedahl. 2019. Big imagery as a resource to understand patterns, dynamics, and vulnerability of Arctic polygonal tundra. Page C13E-1374 in *AGU Fall Meeting Abstracts*, held in Location. Edited by J. Editor. Location: Publisher.

Blaschke, T. 2010. Object based image analysis for remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing* 65(1):2–16.

Blaschke, T., G. J. Hay, M. Kelly, S. Lang, P. Hofmann, E. Addink, R. Q. Feitosa, F. van der Meer, H. van der Werff, F. van Coillie and D. Tiede. 2014. Geographic object-based image analysis—Towards a new paradigm. *ISPRS Journal of Photogrammetry and Remote Sensing* 87:180–191.

Brown, J., O. J. Ferrians Jr., J. A. Heginbottom and E. S. Melnikov. 1997. *Circum-Arctic Map of Permafrost and Ground-Ice Conditions*. Location: US Geological Survey.

Brown, J., O. Ferrians, J. A. Heginbottom and E. Melnikov. 2002. *Circum-Arctic Map of Permafrost and Ground-Ice Conditions, Version 2*. Boulder, Colo.: National Snow and Ice Data Center.

Chi, M., A. Plaza, J. A. Benediktsson, Z. Sun, J. Shen and Y. Zhu. 2016. Big data for remote sensing: Challenges and opportunities. *Proceedings of the IEEE* 104(11):2207–2219. https://doi.org/10.1109/JPROC.2016.2598228.

El-Baz, A. and J. S. Suri, eds. 2020. *Big Data in Multimodal Medical Imaging*. Boca Raton, Fla.: CRC Press.

He, K., X. Zhang, S. Ren and J. Sun. 2016. Deep residual learning for image recognition. Pages 770–778 in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, held in Las Vegas, Nev., 27–30 June 2016. Location: IEEE.

Huerta, E. A., R. Haas, S. Jha, M. Neubauer and D. S. Katz. 2019. Supporting high-performance and high-throughput computing for experimental science. *Computing and Software for Big Science* 3:5.

Jones, B. M., G. Grosse, C. D. Arp, E. Miller, L. Liu, D. J. Hayes and C. F. Larsen. 2015. Recent Arctic tundra fire initiates widespread thermokarst development. *Scientific Reports* 5:15865.

Jorgenson, M. T., Y. L. Shur and E. R. Pullman. 2006. Abrupt increase in permafrost degradation in Arctic Alaska. *Geophysical Research Letters* 33(2):L02503.

Kaiser, S., G. Grosse, J. Boike and M. Langer. 2021. Monitoring the transformation of Arctic landscapes: Automated shoreline change detection of lakes using very high resolution imagery. *Remote Sensing* 13(14):2802.

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

Kanevskiy, M., Y. Shur, M. T. Jorgenson, C.-L. Ping, G. J. Michaelson, D. Fortier, E. Stephani, M. Dillon and V. Tumskoy. 2013. Ground ice in the upper permafrost of the Beaufort Sea coast of Alaska. *Cold Regions Science and Technology* 85:56–70.

Karmas, A., A. Tzotsos and K. Karantzalos. 2016. Geospatial big data for environmental and agricultural application. In *Big Data Concepts, Theories, and Applications*, edited by S. Yu and S. Guo, 353–389. Cham, Switzerland: Springer.

Kokelj, S. V., J. Tunnicliffe, D. Lacelle, T. C. Lantz, K. S. Chin and R. Fraser. 2015. Increased precipitation drives mega slump development and destabilization of ice-rich permafrost terrain, northwestern Canada. *Global and Planetary Change* 129:56–68.

Kremer, J., K. Stensbo-Smidt, F. Gieseke, K. S. Pedersen and C. Igel. 2017. Big universe, big data: Machine learning and image analysis for astronomy. *IEEE Intelligent Systems* 32(2):16–22.

Kucuk, A., J. M. Banda and R. A. Angryk. 2017. A large-scale solar dynamics observatory image dataset for computer vision applications. *Scientific Data* 4:170096.

Lang, S., A. Baraldi, D. Tiede, G. Hay and T. Blaschke. 2018. Towards a (GE) OBIA 2.0 manifesto—Achievements and open challenges in information & knowledge extraction from big Earth data. Pages 18–22 in *Proceedings of the GEOBIA 2018*, held in Montpellier, France, 18–22 June 2018. Edited by J. Editor. Location: Publisher.

LeCun, Y., Y. Bengio and G. Hinton. 2015. Deep learning. *Nature* 521:436–444.

Lee, C. A., S. D. Gasster, A. Plaza, C.-I. Chang and B. Huang. 2011. Recent developments in high performance computing for remote sensing: A review. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 4(3):508–527. https://doi.org/10.1109/JSTARS.2011.2162643.

Liljedahl, A. K., J. Boike, R. P. Daanen, A. N. Fedorov, G. V. Frost, G. Grosse, L. D. Hinzman, Y. Iijma, J. C. Jorgenson, N. Matveyeva, M. Necsoiu, M. K. Raynolds, V. E. Romanovsky, J. Schulla, K. D. Tape, D. A. Walker, C. J. Wilson, H. Yabuki and D. Zona. 2016. Pan-Arctic ice-wedge degradation in warming permafrost and its influence on tundra hydrology. *Nature Geoscience* 9:312–318.

Liu, J., D. Feld, Y. Xue, J. Garcke, T. Soddemann and P. 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.

Liu, P. 2015. A survey of remote-sensing big data. *Frontiers in Environmental Science* 3:45.

Liu, P., L. Di, Q. Du and L. Wang. 2018. Remote sensing big data: Theory, methods and applications. *Remote Sensing* 10(5):711.

Ma, L., Y. Liu, X. Zhang, Y. Ye, G. Yin and B. A. Johnson. 2019. Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS Journal of Photogrammetry and Remote Sensing* 152:166–177.

Ma, Y., H. Wu, L. Wang, B. Huang, R. Ranjan, A. Zomaya and W. Jie. 2015. Remote sensing big data computing: Challenges and opportunities. *Future Generation Computer Systems* 51:47–60.

Muster, S., B. Heim, A. Abnizova and J. Boike. 2013. Water body distributions across scales: A remote sensing based comparison of three Arctic tundra wetlands. *Remote Sensing* 5(4):1498–1523.

Muster, S., M. Langer, B. Heim, S. Westermann and J. Boike. 2012. Land cover classification of Samoylov Island and Landsat subpixel water cover of Lena River Delta, Siberia, with links to ESRI grid files. *Pangaea*, https://doi.org/10.1594/PANGAEA.786927.

Nikolaou, C., K. Kyzirakos, K. Bereta, K. Dogani, S. Giannakopoulou, P. Smeros, G. Garbis, M. Koubarakis, D. E. Molina, O. C. Dumitru, G. Schwarz and M. Datcu. 2014. Big, linked and open data: Applications in the German Aerospace Center. Pages 444–449 in *The Semantic Web: ESWC 2014 Satellite Events*. Edited by V. Presutti, E. Blomqvist, R. Troncy, H. Sack, I. Papadakis and A. Tordai. European Semantic Web Conference 2014, held in Anissaras, Greece, 25–29 May 2014. Lecture Notes in Computer Science vol. 8798. Cham, Switzerland: Springer.

Nitze, I., G. Grosse, B. M. Jones, V. E. Romanovsky and J. Boike. 2018. Remote sensing quantifies widespread abundance of permafrost region disturbances across the Arctic and Subarctic. *Nature Communications* 9:5423.

Okunev, A. G., M. Y. Mashukov, A. V. Nartova and A. V. Matveev. 2020. Nanoparticle recognition on scanning probe microscopy images using computer vision and deep learning. *Nanomaterials* 10(7):1285.

Raynolds, M. K., J. C. Jorgenson, M. T. Jorgenson, M. Kanevskiy, A. K. Liljedahl, M. Nolan, M. Sturm and D. A. Walker. 2020. Landscape impacts of 3D-seismic surveys in the Arctic National Wildlife Refuge, Alaska. *Ecological Applications* 30(7):e02143.

Raynolds, M. K., D. A. Walker, A. Balser, C. Bay, M. Campbell, M. M. Cherosov, F.J.A. Daniëls, P. B. Eidesen, K. A. Ermokhina, G. V. Frost, B. Jedrzejek, M. T. Jorgenson, B. E. Kennedy, S. S. Kholod, I. A. Lavrinenko, O. V. Lavrinenko, B. Magnússon, N. V. Matveyeva, S. Metúsalemsson, L. Nilsen, I. Olthof, I. N. Pospelov, E. B. Pospelova, D. Pouliot, V. Razzhivin, G. Schaepman-Strub, J. Šibík, M. Y. Telyatnikov and E. Troeva. 2019. A raster version of the Circumpolar Arctic Vegetation Map (CAVM). *Remote Sensing of Environment* 232:111297.

Schmied, C., P. Steinbach, T. Pietzsch, S. Preibisch and P. Tomancak. 2016. An automated workflow for parallel processing of large multiview SPIM recordings. *Bioinformatics* 32(7):1112–1114.

Skurikhin, A. N., C. J. Wilson, A. Liljedahl and J. C. Rowland. 2014. Recursive active contours for hierarchical segmentation of wetlands in high-resolution satellite imagery of arctic landscapes. Pages 137–140 in *2014 Southwest Symposium on Image Analysis and Interpretation*, held in San Diego, Calif., 6–8 April 2014. Edited by J. Editor. Location: IEEE.

Steedman, A. E., T. C. Lantz and S. V. Kokelj. 2017. Spatio-temporal variation in high-centre polygons and ice-wedge melt ponds, Tuktoyaktuk Coastlands, Northwest Territories. *Permafrost and Periglacial Processes* 28(1):66–78.

Wang, L., Y. Ma, J. Yan, V. Chang and A. Y. Zomaya. 2018. pipsCloud: High performance cloud computing for remote sensing big data management and processing. *Future Generation Computer Systems* 78(1):353–368.

Wang, L., W. Song and P. Liu. 2016. Link the remote sensing big data to the image features via wavelet transformation. *Cluster Computing* 19:793–810.

Witharana, C., M.A.E. Bhuiyan and A. K. Liljedahl. 2019. Towards first Pan-Arctic ice-wedge polygon map: Understanding the synergies of data fusion and deep learning in automated ice-wedge polygon detection from high-resolution commercial satellite imagery. Page C22C-07 in *AGU Fall Meeting Extracts*, held in San Francisco, Calif., 9–13 December 2019. Edited by J. Editor. Location: Publisher.

Witharana, C., M.A.E. Bhuiyan, A. K. Liljedahl, M. Kanevskiy, T. Jorgenson, B. M. Jones, R. Daanen, H. E. Epstein, C. G. Griffin, K. Kent and M. K. Ward Jones. 2021. An object-based approach for mapping tundra ice-wedge polygon troughs from very high spatial resolution optical satellite imagery. *Remote Sensing* 13(4):558.

Zhang, W., C. Witharana, A. K. Liljedahl and M. Kanevskiy. 2018. Deep convolutional neural networks for automated characterization of Arctic ice-wedge polygons in very high spatial resolution aerial imagery. *Remote Sensing* 10(9):1487.

Assignment of Copyright to ASPRS

Current copyright law requires that authors of papers submitted for publication in *Photogrammetric Engineering & Remote Sensing* transfer copyright ownership to the American Society for Photogrammetry and Remote Sensing before the paper may be published. Upon receipt of this form with your Master Proof, please complete this form and forward it to the Production Coordinator (address below).

Manuscript Number:

Manuscript Title:

Author(s):

Assignment of Copyright in the above-titled work is made on (date)________________________ from the above listed author(s) to the American Society for Photogrammetry and Remote Sensing, publisher of *Photogrammetric Engineering & Remote Sensing*.

> **In consideration of the Publisher's acceptance of the above work for publication, the author or co-author(s) hereby transfer(s) to the American Society for Photogrammetry and Remote Sensing the full and exclusive copyright to the work for all purposes for the duration of the copyright. I (we) understand that such transfer of copyright does not preclude specific personal use, provided that prior to said use, permission is requested from and granted by the American Society for Photogrammetry and Remote Sensing.**
>
> **I (we) acknowledge that this paper has not been previously published, nor is it currently being considered for publication by any other organization.**

____

Author/co-author signatures

Co-authors may fill out and submit this form separately (photocopies are acceptable) if desired; but all co-authors must sign either this or a separate form.

Special Note to U.S. Government Employees

Material prepared by U.S. Government employees as part of their official duties need not have the assignment of copyright transferred since such material is automatically considered as part of the public domain.

If your paper falls within this category please check the appropriate statement and sign below.

__________ **This paper has been prepared wholly as part of my (our) official duties as (a) U.S. Government Employee(s). I (we) acknowledge that this paper has not previously been published, nor is it currently being considered for publication, by any other organization.**

__________ **This paper has been prepared partly in the course of my (our) official duties as (a) U.S. Government Employee(s). For any part(s) not prepared in the course of my (our) official duties, copyright is hereby transferred to the American Society for Photogrammetry and Remote Sensing. I (we) acknowledge that this paper has not previously been published, nor is it currently being considered for publication, by any other organization.**

____

Author/co-author signatures

Please return this form to:
ASPRS
225-408-4422 (fax) or managingeditor@asprs.org