

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/368368729>

ArcCI: A high-resolution aerial image management and processing platform for sea ice

Chapter · January 2023

DOI: 10.1130/2022.2558(06)

CITATION

1

READS

116

10 authors, including:



Dexuan Sha

George Mason University

42 PUBLICATIONS 757 CITATIONS

SEE PROFILE



Anusha Srirenganathan Malarvizhi

George Mason University

7 PUBLICATIONS 46 CITATIONS

SEE PROFILE



Hai Lan

University of Maryland, College Park

29 PUBLICATIONS 602 CITATIONS

SEE PROFILE



Xin Miao

Missouri State University

27 PUBLICATIONS 599 CITATIONS

SEE PROFILE

ArcCI: A high-resolution aerial image management and processing platform for sea ice

Dexuan Sha

Anusha Srirenganathan Malarvizhi

Hai Lan

George Mason University, Geography and Geoinformation Science, 4400 University Drive, Fairfax, Virginia 22030, USA

Xin Miao

*Missouri State University, Department of Geography, Geology and Planning,
901 S. National Avenue, Springfield, Missouri 65897, USA*

Hongie Xie

*The University of Texas at San Antonio, Department of Earth and Planetary Sciences,
One UTSA Circle, San Antonio, Texas 78249, USA*

Daler Khamidov

*George Mason University, Department of Computer Science, Nguyen Engineering Building,
4400 University Drive, Fairfax, Virginia 22030, USA*

Kevin Wang

*University of California, Electrical Engineering & Computer Sciences,
387 Soda Hall #1776, Berkeley, California 94720-1776, USA*

Seren Smith

Smith College, Department of Statistical & Data Sciences, McConnell Hall 214, Northampton, Massachusetts 01063, USA

Katherine Howell

Science Systems & Applications, 10210 Greenbelt Road #600, Lanham, Maryland 20706, USA

Chaowei Yang

George Mason University, Geography and Geoinformation Science, 4400 University Drive, Fairfax, Virginia 22030, USA

ABSTRACT

The Arctic sea-ice region has become an increasingly important study area since it is not only a key driver of the Earth's climate but also a sensitive indicator of climate change. Therefore, it is crucial to extract high-resolution geophysical features of sea ice from remote sensing data to model and validate sea-ice changes. With large volumes of high spatial resolution data and intensive feature extraction, classification,

and analysis processes, cloud infrastructure solutions can support Earth science. One example is the Arctic CyberInfrastructure (ArcCI), which was built to address image management and processing for sea-ice studies. The ArcCI system employs an efficient geophysical feature extraction workflow that is based on the object-based image analysis (OBIA) method alongside an on-demand web service for Arctic cyberinfrastructure. By integrating machine learning classification approaches, the on-demand sea-ice high spatial resolution (HSR) imagery management and processing service and framework allows for the efficient and accurate extraction of geophysical features and the spatiotemporal analysis of sea-ice leads.

1. INTRODUCTION

Polar regions have become an increasingly important research area as they provide significant natural resources, function as sensitive indicators of climate changes, and are a key driver of the Earth's climate. High spatial resolution (HSR) aerial imagery can provide critical information for better understanding, utilizing, and protecting polar regions. To effectively and efficiently collect, manage, and process large amounts of HSR images, a polar cyberinfrastructure (CI) is necessary. To increase our understanding of fragile polar environments and facilitate critical decision-making, such a CI needs to be capable of aiding researchers in collecting and integrating heterogeneous image data, extracting spatiotemporal patterns of sea ice, and linking sea-ice features to the surrounding dynamics and, in particular, to thermodynamic phenomena.

In the past few years, the amount of HSR aerial images collected and processed has increased dramatically alongside expansions in data collection platforms, storage capacity, and computational power. For example, unmanned aerial vehicle technology has greatly expanded the ability to collect HSR images for land-cover use classification, environmental monitoring, and natural resource mapping. (Sawant and Mohite, 2018; Bühler et al., 2016; Seier et al., 2017). In polar science, HSR imagery provides more detail in the spatial dimension, making the sea-ice features easily identifiable. For example, sea-ice leads are elongated cracks in the sea ice that develop due to the diverging or shearing of floating ice floes as they move with currents and wind (Wang, et al., 2016). Ice leads ranging from 1 m to 100 m are not discernible in a 25 km satellite image but are visible in an HSR aerial photo with 1 m spatial resolution. These HSR images (with 0.05 m to 1 m resolution) usually require a lot of storage space and efficient processing procedures (Nishar et al., 2016; Bühler et al., 2016). Most projects only use local storage systems or servers to archive and process HSR images, but Li et al. (2015) discuss the various procedures necessary for transitioning from local to distributed storage systems for long-term data collection. Amazon Web Services (AWS) and Google Earth Engine (GEE) have been introduced for scalable and efficient cloud storage, as well as for computationally intensive deep learning (DL) image processing algorithms (Ampatzidis et al., 2020; Tamiminia et al., 2020).

Polar domain-specific CI is important for the following reasons: (1) considering geospatial principles (such as spatial constraints and feature relationships) specific to the polar region; (2) supporting sophisticated data management, storage, and visualization for the polar region (for example, polar-focused projections); and (3) supporting geospatial modeling that provides insight into the past, present, and future state of the polar regions (Yang et al., 2010). In the past decade, Polar CIs have evolved considerably. The first-generation polar CI consisted of static data infrastructure, with a focus on data-level interoperability, and only provided data storage and portals. For example, the Arctic Research Mapping Application was designed to access, query, and browse the Arctic Research Logistics Support Service database (Walker Johnson et al., 2011). The first-generation CI mainly served as a data archive, providing data deposits only in static web pages. The second-generation CI started to consider active and intelligent data discovery and access through web crawlers and internet mining (Li et al., 2017; Mattmann, 2013; Jiang et al., 2018). The current third generation of CIs, referred to as data gateways (Sha et al., 2020), provides much more advanced data integration functionalities and visualization approaches but still lacks publicly available image exploration tools that advance knowledge-based decision making. Currently, the emerging fourth generation of CIs can be defined as a knowledge infrastructure that provides interactive analysis and reasoning modules. Examples that have been developed include a multi-faceted visualization module for complex climate patterns and an intelligent spatiotemporal reasoning system (Li et al., 2015; Jiang et al., 2017).

Furthermore, the general-purpose platforms such as GEE can support abundant analysis functionalities with customized application programming interfaces (APIs). However, GEE has challenges in the design and use of the system. The challenges are (1) a limitation of computing resources to ensure that the users do not take over control of the shared resources, (2) poor performance for operations in which the cell value depends on the arbitrarily neighboring cells such as classical clustering algorithms, and (3) the user's unfamiliarity with the underlying client/server programming model (Gorelick et al., 2017).

Besides the above-mentioned cloud-based GEE implementation, public cloud computing techniques have made possible large-scale computing operations such as massive parallel

simulations and satellite image processing. However, in the past, cloud computing has significantly decreased efficiency due to two factors: (1) the absence of high bandwidth and (2) low-latency connection with virtual machines (Yelick et al., 2011). To overcome the above-mentioned factors, cloud-based systems require high-performance networks and improved communication between nodes for message passing interface (MPI) libraries. MPI is a long-established communication protocol that is designed to support parallel programming (Zhuang et al., 2020). Recent improvements in AWS have allowed for “near-bare metal” performance for virtual machine management, a new C5n instance (C5n, Amazon, 2022) in AWS 100 Gb/s bandwidth (Amazon, 2018), and a new low-latency network interface called Elastic Fabric Adapter that improves communication for MPI nodes (Zhuang et al., 2020). The efficient performance and accessibility of the AWS cloud software have allowed for satellite image data to be processed and stored in the cloud, reducing time and costs for hardware setup and management.

As HSR images show more spatial details, they require more disk space to store petabytes of information. Therefore, cloud storage services are more suitable than local storage because they provide highly scalable and reliable storage services. To efficiently store, process, and retrieve data, different storage locations can be assigned to disparate data sets, e.g., to centralize the metadata from the HSR images for management while providing the efficient storage and parallel computing capabilities of the cloud platform with distributed storage (Zheng et al., 2018). HSR images usually require pre-processing operations such as geometric and radiometric correction. Therefore, parallel computing may play a significant role in performing these operations. Kulawiak and Chybicki (2018) reported that utilizing hyperthreading, a hardware setting that allows more than one thread to run on each core (Intel, 2022), leads to reduced execution time for geospatial data processing. However, it is worth noting that latency issues in cloud environments were not considered and could be a potential factor in determining the efficiency of cloud storage depending on workload amount. The flexibility of cloud storage enables the utilization of software like ArcGIS to store satellite images in an optimal fashion and run spatial analysis modules to provide a web service (Huang et al., 2018). Furthermore, making satellite images available through web services allows more users to explore the data for comparative studies.

Currently, therefore, there is no highly specialized Arctic CI building block that offers (1) HSR sea-ice image collection, (2) on-demand value-added services like automatic batch image classification and physical parameter extraction, and (3) interactive spatiotemporal analysis of sea-ice evolution. Accordingly, the motivation for this project was to develop a module that can serve both the Arctic Sea-ice community and the larger polar science community. Specifically, this project aimed to classify the HSR aerial imageries into four sea-ice types: thick ice, thin ice, shadow, and water. The classification was implemented using a machine learning (ML)-based image processing mod-

ule called Open Source Sea-Ice Processing (OSSP) (Wright and Polashenski, 2018).

This CI uses examples of sea-ice classification obtained from the Operation IceBridge digital mapping system (DMS) and is designed to upload, read, and classify images with an example of DMS Level-1B geolocated and orthorectified images in GeoTIFF (TIF) format with associated metadata. The classification of sea-ice physical parameters can be applied to address scientific objectives such as, but not limited to, (1) analyzing the evolution of ice concentration and edge, size distributions of floes, melt pond distributions, lateral melting processes, surface roughness, and ridge heights; (2) examining the air-ocean heat transfer through leads/water, melt ponds, submerged ice, and bare and snow-covered ice; (3) examining fresh water volume and change based on melt pond distribution, depth, and areas; and (4) calibrating and validating sea-ice modeling output and parameters (Sha, 2021).

Given the challenges of big data and the lack of customized polar CI and web services, this research aimed to create a comprehensive image management and processing platform called ArcCI that includes image-data lifecycle functions for loading, storage, sharing, processing, result validation, and analysis. Creating a public, cloud-based platform enabled high-performance computing that allows for massive image processing requests from multiple users. To show the effectiveness of the cloud computing platform, we conducted performance experiments in terms of batch processing duration and central processing unit (CPU) utilization. The platform also included a DL benchmark for sea-ice image classification. The functional components of the ArcCI include (1) image management to upload, view, search, share, and delete HSR images; (2) user management; (3) image analysis function; (4) image batch processing; and (5) map visualization.

2. CLOUD-BASED ARCHITECTURE AND AWS COMPONENTS

The ArcCI architecture is illustrated in Figure 1. From bottom to top, it consists of three layers: software layer, service layer, and application layer.

The fundamental layer is the configured software layer (Layer 1) that includes the operating system, cloud software, and database management system to provide on-demand, elastic, and cloud services. The software layer consists of the AWS cloud computing environment, and capability integration is conducted to best leverage the cloud computing environment for polar sciences. The cloud components include (1) AWS Elastic Beanstalk, a service for deploying and scaling web applications (Amazon, 2022a); (2) Amazon Elastic Compute Cloud (Amazon EC2), a service that provides secure and reliable computing capacity in the cloud (Amazon, 2022b); (3) AWS Lambda, a serverless, event-driven computing service that allows users to run applications virtually (Amazon, 2022c); (4) Amazon Relational Database Service (RDS), a service to set up, operate, and scale relational databases in the cloud (Amazon, 2022d); and (5) Amazon

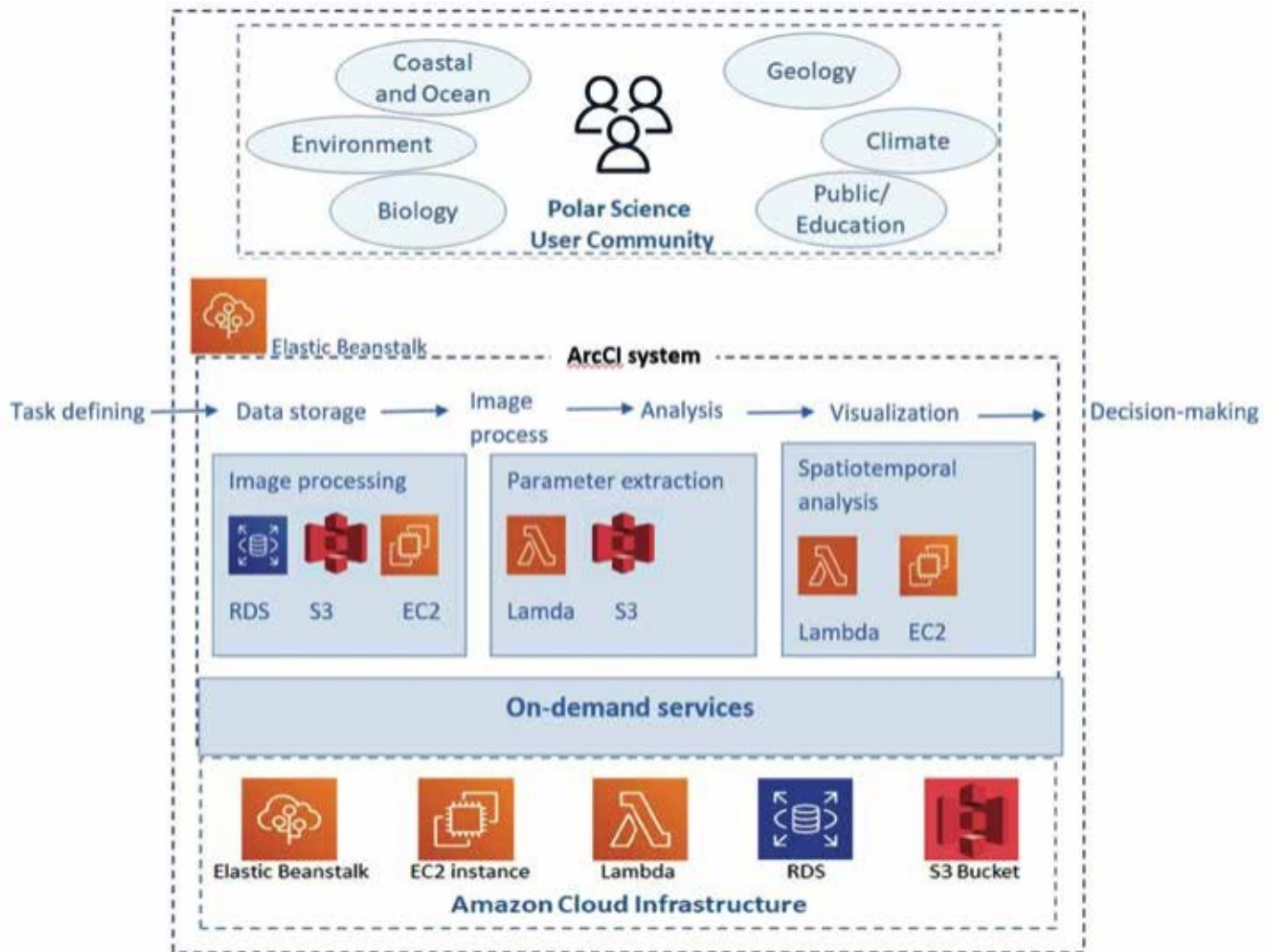


Figure 1. Conceptual cloud-based architecture of the ArcCI system consists of three layers from bottom to top: the software layer, service layer, and application layer.

Simple Storage Service (Amazon S3), an object-storing service that offers high scalability and reliability (Amazon, 2022e). All of the above-mentioned services can also be implemented in George Mason University's community cloud computing environment (Yang et al., 2011, 2013).

Layer 2, developed through this project, provides different types of on-demand services, including image processing, parameter extraction, and spatiotemporal visual analyses, among others. This layer provides a graphical user interface (UI) to be integrated based on our research and will install on desktop computers or mobile computing devices (Gui et al., 2013a, 2013b) to support the data life cycle of generation/discovery, processing, analysis, and visualization for end-users (Li et al., 2011).

The top layer is the application layer (Layer 3), which can be customized by end-users according to their polar science research needs. For example, the users can customize the application layer based on their study areas (Arctic or Antarctica), image process-

ing methods, and visualization techniques. To better support image analysis and polar science research, relevant middleware in the cloud environment could be integrated to allow the ArcCI to address polar science data processing and sharing challenges.

The five essential cloud-based AWS services are explained below.

2.1. Cloud-Based Distributed File System Using S3

ArcCI is designed to host big data from multiple agencies and polar scientists. A backup distributed file system (DFS) and synchronized storage is provided in the ArcCI system for the polar science community. The DFS provides transparent replication and fault tolerance to enhance reliability. The backup storage automatically makes a secondary copy (or even additional copies) of the data that is available for recovery if the original data are damaged (Yang et al., 2013, Chapter 3). The synchronization

enables users to access the same copy of data from multiple virtual machines across AWS regions. To minimize data transfer, data transformation and allocation are optimized based on the volume of data, user distribution, network configuration, and the patterns of backup resource utilization in space and time (Li et al., 2017). Such optimization considers the geographic location of data users and the temporal patterns of their access requirements. Therefore, data are allocated closely to data users and synchronized for data consistency in the cloud-distributed physical infrastructure across the world (Yang et al., 2013, Chapter 11).

Since Elastic Block Store (EBS) volumes will be deleted when we terminate the EC2 instance, we use S3 for persistent storage (Zhuang et al., 2019). S3 storage is independent of EC2 and can be shared across distributed computing nodes. To ease the transfer and retrieval of sea-ice images, we mount S3 to EC2 using Rclone along with winfsp (RCLONE, 2022). The data transfer between EC2 and S3 happens seamlessly without requiring users to transfer explicitly. Each ArcCI user has his or her own folder (parent folder) containing the sea-ice images he or she uploaded. This folder management system, which is embedded in S3, ensures data integrity and security.

2.2. Beanstalk for Front-End Interface and Load Balancing

To enable the auto-scaling, load balancing, and scheduling of the tasks running on ArcCI, the AWS Beanstalk component is utilized to deploy applications in the cloud easily and quickly (Bellenger et al., 2011). Hypertext Preprocessor (PHP) software was used to develop the web interface that can be automatically deployed to AWS using Beanstalk. In addition to deployment, Beanstalk handles load balancing, autoscaling, and application health checking. For future enhancement, we will use load balancer to distribute the incoming traffic across multiple instances. This middleware function enables the system to monitor the status of all tasks currently running on ArcCI as well as the workload of all virtual machines provisioned by ArcCI.

2.3. EC2 for Elastic Instance of OSSP

ArcCI utilizes AWS EC2 to host virtual machines running the Windows Server 2019 Operating System (OS). Using the AWS console, each EC2 instance is configured with appropriate CPU and RAM. AWS enables users to monitor the performance metrics (CPU, disk utilization, and network bandwidth) of EC2 instances. We use these metrics to either upscale or downscale the AWS instance manually.

2.4. Lambda for DL Classification Function

AWS SageMaker (Amazon, 2022f) and AWS API Gateway (Amazon, 2022g) are used alongside AWS Lambda to deploy our pre-trained model, DeepLabV3 (Chen et al., 2017). In section 3.2 we explain this pre-trained model in detail. AWS Sagemaker

is a cloud ML platform that provides developers with the ability to create, train, and deploy ML models. AWS API Gateway provides developers with the ability to create, publish, maintain, monitor, and secure APIs, while AWS Lambda allows developers to run code in response to events. Our model is deployed to SageMaker, where a model endpoint for production is created. API Gateway handles hosting, and HTTP requests are caught by a designated Lambda function that is invoked after it hits the API Gateway. The Lambda function verifies incoming data, calls the SageMaker endpoint, and returns the correct response. Since the size limit for API Gateway may be exceeded, the classified images are uploaded to an S3 bucket. Then the Lambda function will retrieve the image from the bucket and invoke the model.

2.5. Relational Database Management System for Metadata and Business Data

ArcCI utilizes the Amazon RDS, which can easily set up, configure, and scale relational databases in the cloud. Using the AWS console, we provisioned a MySQL database and completed initial configuration settings. The database design for ArcCI web application includes tables, indexes, and constraints. The image attribute table is one of the major tables that stores metadata related to an HSR image. During image upload, information such as file path, upload time, status, and upload username are stored in the table. Ancillary spatial information such as latitude, longitude, and altitude, along with shuttle (pitch and roll) and photographic (shutter speed and f-stop) information are also stored in the image table.

3. ML/DL-BASED IMAGE CLASSIFICATION FOR HSR AERIAL SURVEY DATA

3.1. OSSP and Parallel Computing

High spatial resolution image processing is the major feature of ArcCI. Historically, most of the high-resolution sea-ice aerial or ship-based photos were analyzed through pixel-based methods (Lu et al., 2010; Renner et al., 2013; Jiang et al., 2017). Pixel-based methods based on pixel brightness values or spectral values ignore spatial autocorrelation and generate “salt-and-pepper” noise in classification (Liu and Xia, 2010; Xie et al., 2007). In contrast, object-based classification is based on image segmentation, the process of partitioning an image into multiple objects or groups of pixels, which makes classifications more meaningful and easier to analyze (Hussain et al., 2013; Shapiro and Stockman, 2001). This method not only considers spectral values but also spatial measurements that characterize the shape, texture, and contextual properties of the region so as to potentially improve classification accuracy (Liu and Xia, 2010). Figure 2 demonstrates the three major steps of the algorithm, including (1) object-based image segmentation, which converts neighboring pixels into a large object as the classification unit; (2) a feature engineering process by which reasonable object-based features

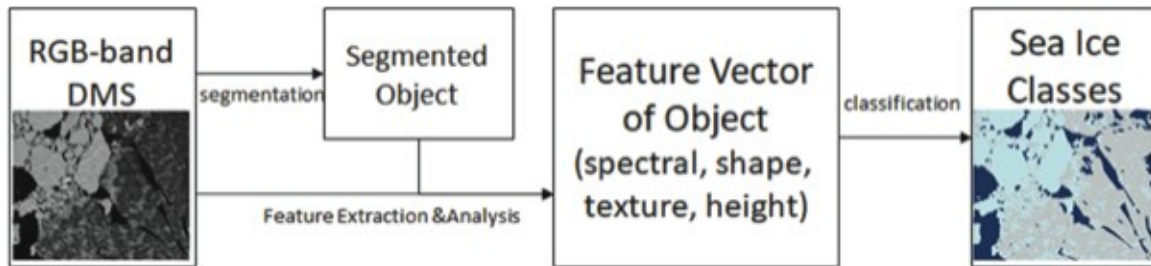


Figure 2. Image processing flow chart is based on the object-based image analysis and machine learning methods. RGB-band image courtesy of the National Snow and Ice Data Center, University of Colorado, Boulder.

of each sea-ice class are extracted; and (3) a supervised ML classifier to label the class of each spatial object. This ML image processing module was programmed using the OSSP Python library (Wright and Polashenski, 2018), and the package is integrated into EC2 images as an on-demand instance service. To speed up the batch processing workflow, the customized parallel computing mode was implemented in OSSP using a divide-and-conquer strategy. In the single image process, the whole input HSR image is divided into several sub-images to be segmented and classified separately, and the classified results are merged back by the default spatial distribution of the divided subsets. This allows each of the subtasks to be assigned to multiple CPU cores in parallel to achieve a high-performance, single image process.

Since Arctic sea-ice image processing is usually not time-sensitive, we believe that this process is affordable in terms of computation and transfer burdens. Furthermore, we provide two options: (1) researchers can send us their raw images, and we will upload and publish the image and processed results through ArcCI; or (2) researchers can upload their raw images for service only, and we will release a copy of the processed results. Through the latter method, the extracted information (sea-ice features and physical parameters as vector layers) can be shared through the Internet more efficiently. The image data, extracted features, and process are released in two ways, through (1) Open Geospatial

Consortium (OGC)-compliant web services, such as Web Map Service (Open Geospatial Consortium, 2022c), Web Coverage Service (Open Geospatial Consortium, 2022a), and Web Feature Service (Open Geospatial Consortium, 2022b), which can be easily integrated with virtual globes, such as Google Earth, to provide a straightforward spatiotemporal visualization approach; and through (2) on-demand service (in compliance with OGC Web Processing Service) for end users to leverage and process their own polar images.

3.2. DL Model

Semantic image segmentation is a fundamental computer vision task in which parts of an image belonging to the same object class are clustered together in the form of pixel-level prediction. It has been applied to multiple use cases in the field of remote sensing, including the classification of HSR imagery. Within the past decade, tremendous efforts to advance pixel-level accuracy have led to the emergence of new DL methodologies that have improved the performance of data sets such as Cityscapes and PASCAL VOC (Yuan et al., 2021). These DL methodologies have demonstrated superior performance and success in semantic segmentation as they automatically derive features tailored for targeted classification tasks and allow for

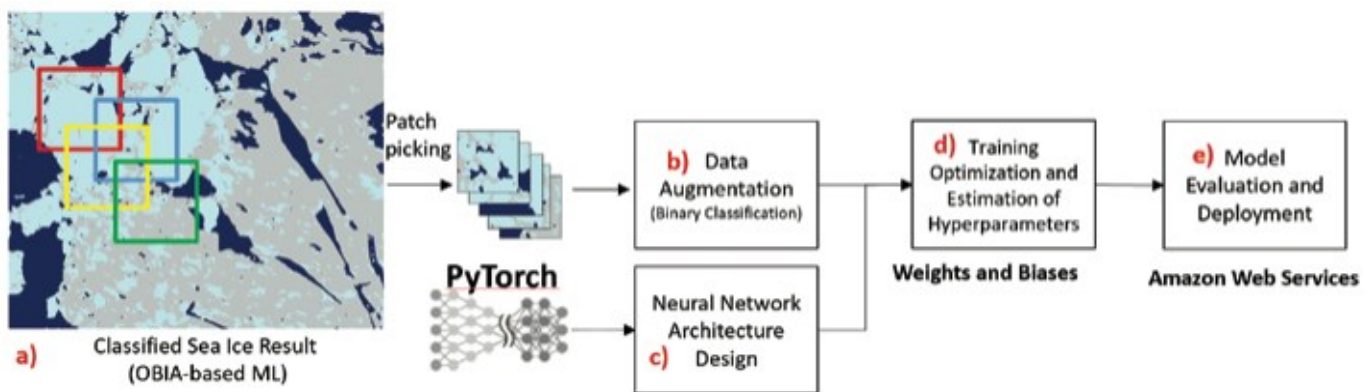


Figure 3. Deep learning semantic segmentation pipeline for sea-ice classification is shown. OBIA—object-based image analysis; ML—machine learning.

improved performance in complex scenarios. The same improvements in performance and success that DL methodologies have enabled in other semantic segmentation applications can also be applied to the classification of sea-ice types. Hence, we developed and integrated a DL model pipeline into the ArcCI platform for the accurate classification of sea-ice types.

The DL semantic segmentation pipeline is as follows:

- (1) The pipeline begins with a data preprocessing stage where the `albumentations` Python package is employed to select 256×256 patches from NASA Level-1B (L-1B) DMS HSR imagery labeled with OBIA ML (Fig. 3), enabling us to gather/create thousands of training images from 8 to 20 HSR images.
- (2) The data preprocessing stage also includes a binary classification script developed for lighting adjustment so that darker images will be easier for the model to process.
- (3) The data preprocessing stage is followed by training. PyTorch (Paszke et al., 2019) is utilized as the main DL framework alongside PyTorch Lightning (Lightning, 2022), a high-level interface for PyTorch built for researchers that allows for the easy logging of metrics, profiling, and distributed training.
- (4) During the training process, the model is evaluated and hyperparameter tuning is conducted using packages such as `Torchmetrics` (Torchmetrics, 2022) and `Weights and Biases (W&B)` (Biewald, 2020). W&B allows for more efficient hyperparameter tuning through the running of sweeps, which tests hundreds of different hyperparameter combinations and displays results for rapid iteration on model performance improvement.
- (5) Since the ArcCI platform is hosted on AWS Lambda, we plan to take advantage of the full suite of ML solutions gathered under the umbrella of AWS when we integrate the DL into the platform.

4. IMPLEMENTATION AND PERFORMANCE TESTING

The ArcCI system was implemented to support the web-based geoscience information services and dynamic interaction for end-users. Web development technologies such as Hypertext Markup Language 5 (HTML 5), JavaScript, and Asynchronous JavaScript and XML (AJAX) calls were used to develop interactive, light-weight, user-friendly, and rich interface web pages. We leveraged the above-mentioned technologies for ArcCI development. HTML 5 defines the structure and presentation of the web page; JavaScript is mainly used for client-side validation, sending user notifications, and designing interactive web pages; and AJAX calls are used to send or receive data from the server without refreshing the entire page.

For server-side development, the PHP was used, which is an open-source scripting language to develop interactive web pages (PHP, 2022). The PHP scripts can seamlessly be embedded into HTML pages that will be executed each time the page is loaded. WAMP is an acronym for Windows, Apache, MySQL, and PHP

(WampServer, 2022). It is a software stack, which means that installing WAMP automatically installs Apache, MySQL, and PHP for Windows server. Apache is a web server that receives user requests from the browser and responds back with the relevant information in web pages. For spatial data management, storage, and retrieval, PostgreSQL was used. This powerful relational database has useful features such as data integrity checking, reliability, disaster recovery, security, extensibility (supports spatial extension using PostGIS), and concurrency (PostgreSQL, 2022).

4.1. Implementation of All Functions

Figure 4 shows the major functional components of the ArcCI. They are (1) image management to upload, view, search, share, and delete HSR images, (2) user management, (3) image analysis function, (4) image batch processing, and (5) map visualization. The components were implemented using the technologies mentioned at the beginning of this section.

4.1.1. Image Management

Image upload: The ArcCI system allows users to perform image input/output operations. Currently, users can upload only TIF images from the IceBridge DMS L-1B Geolocated and Orthorectified Images data set consisting of Level-1B imagery taken from the DMS over the Arctic and Antarctica. The system supports multiple file uploads based on user privilege. During the image upload, metadata such as acquisition date, altitude, latitude, and longitude are retrieved and stored in the database. The actual image is loaded into the DFS (S3). To ensure security and privacy, the file management is organized and managed so that images are not made visible to other users.

Image compression: The original HSR images are several megabytes, which makes them difficult to render in the UI for visualization. Thumbnail images are reduced versions of the original images. PHP Imagick Library is used to compress the image while maintaining the aspect ratio of the original image.

Image view: Users can view the HSR images. To render the image, the web client makes an XMLHttpRequest to the web server. This helps to load the massive amount of data without reloading the whole page.

Image share: ArcCI offers a user-friendly interface for the image owners/uploader to select specific users or all users in the system with whom to share images. Users are only allowed to view the shared images. They can neither process nor delete them.

Image search, delete, and download: Users have the option to search by username or image name. The database design includes table indexing to optimize the search function. Additionally, users can delete images uploaded by them and download the original and classified images to their local machines.

4.1.2. User Management

The ArcCI gateway enables users to register accounts to upload and manage images. User management features are (1) a user authentication process to verify the registered email, (2) session

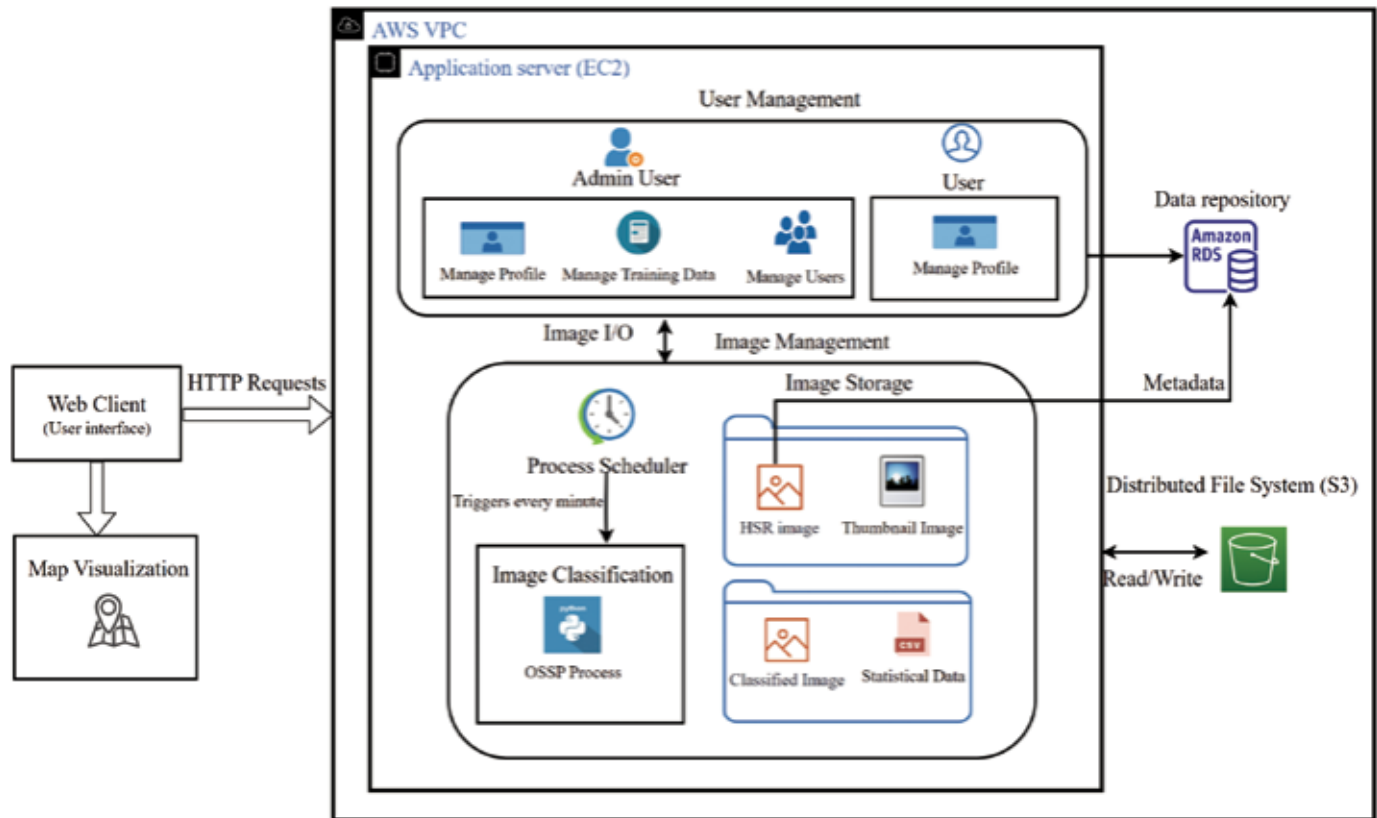


Figure 4. Functional components of the ArcCI system with user interaction, data repository, and distributed file system are shown.

management that securely handles and manages requests from a single user, and (3) user access-level management. Each user is assigned one of three levels, namely General, Privileged, and Administrator. Table 1 shows the user levels and their respective image processing privileges. Users with administrator privileges can manage users and training data sets and assign user levels to others. Additionally, a “Default” user uploads both processed and unprocessed images for others to explore.

4.1.3. Image Classification Analysis, and Display Function

The ArcCI system provides a classification tool that allows users to select parameters required by the OSSP process. The parameters include a segmentation function, a training data set, feature selection, and a machine classifier. The OSSP process detects the geophysical parameters and their variations. The sea-ice classification scheme consists of four classes: narrow open water, thin ice, thick ice, and shadow. After the completion of

classification, the user can visualize the raw HSR image and classified image side by side (Fig. 5). The result of the classification can also be visualized in a responsive, cross-browser-compatible pie chart (Fig. 6).

4.1.4. Image Batch Processing

To reduce the burden on computing resources, we implemented image batch processing. The batch processing framework (Fig. 6) consists of (1) an image database to store HSR images selected by users for classification; (2) a process scheduler, triggered every minute, to submit images for processing; and (3) an OSSP task handler to monitor and manage the images being processed. First, when a user or multiple users submit images for processing, the image batch table stores the submission time and processing status. Second, the process scheduler creates the job queuing process on a first-come, first-served basis and submits the images. Every minute it searches for new images to process

TABLE 1. DIFFERENT USER LEVELS AND IMAGE PROCESSING PRIVILEGES

User level	Max file upload	Max file size	Max ongoing file processes	Max processing thread
General	1	100	1	1
Privileged	10	200	5	4
Administrator	20	1000	30	8

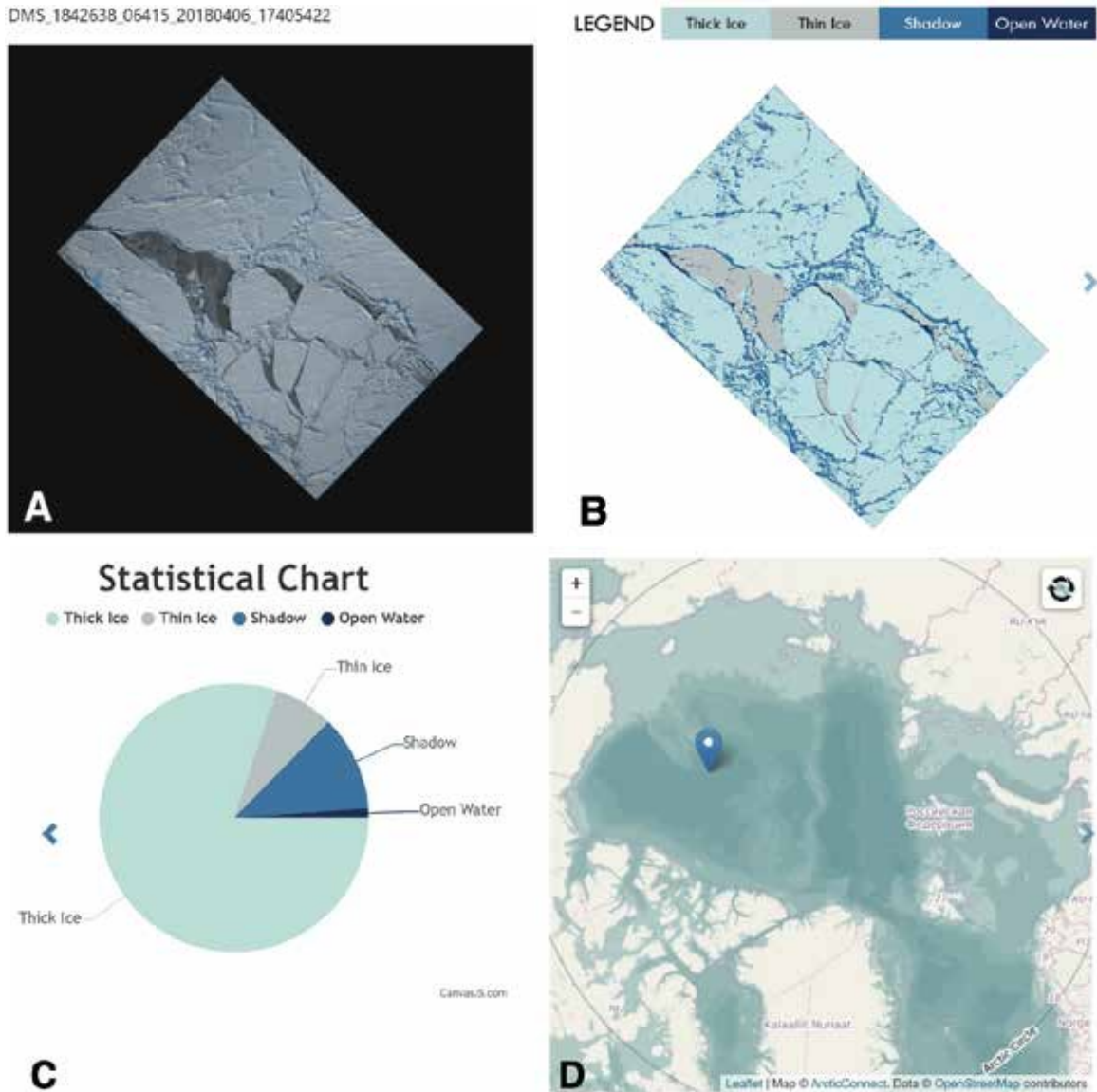


Figure 5. (A) Original digital mapping system (DMS) image is shown. (B) The same image is classified into four sea-ice types. (C) Statistical chart of classified result. (D) Image location of the DMS image. Figures 5A and 5B courtesy of the National Snow and Ice Data Center, University of Colorado, Boulder. Figure 5D courtesy ArcticConnect; map data copyrighted by PolarMapJS and by OpenStreetMap contributors under the Open Database License and available from https://webmap.arcticconnect.ca/index.html#ac_3573/2/90.0/0.0.

in the batch table. Third, the OSSP task handler determines the number of images that can be processed at a time and starts the OSSP process. The handler monitors the change in image status when the process is completed and processes the next image in the queue.

4.1.5. Map Visualization

Figure 7 shows the map visualization tool implemented in ArcCI. The visualization was implemented using Arctic Web Map (AWM), an Arctic-focused web mapping tool that offers customized map projections specific to the Arctic region (AWM, 2022).

AWM has two components: (1) tile server, and (2) PolarMap.js, a Leaflet-based JavaScript library for interactive mapping (Leaflet, 2022). The current AWM tiles support six projections, namely EPSG:3571, EPSG:3574, EPSG:3572, EPSG:3573, EPSG:3575, and EPSG:3576.

Additionally, the visualization tool offers a responsive and interactive graphical UI for exploring, visualizing, and analyzing sea ice. The visualization allows the user to zoom in/out, pan, and filter the image based on its metadata. The filter tool enables users to search images based on various parameters, namely image acquisition date, uploaded users, and image process status.

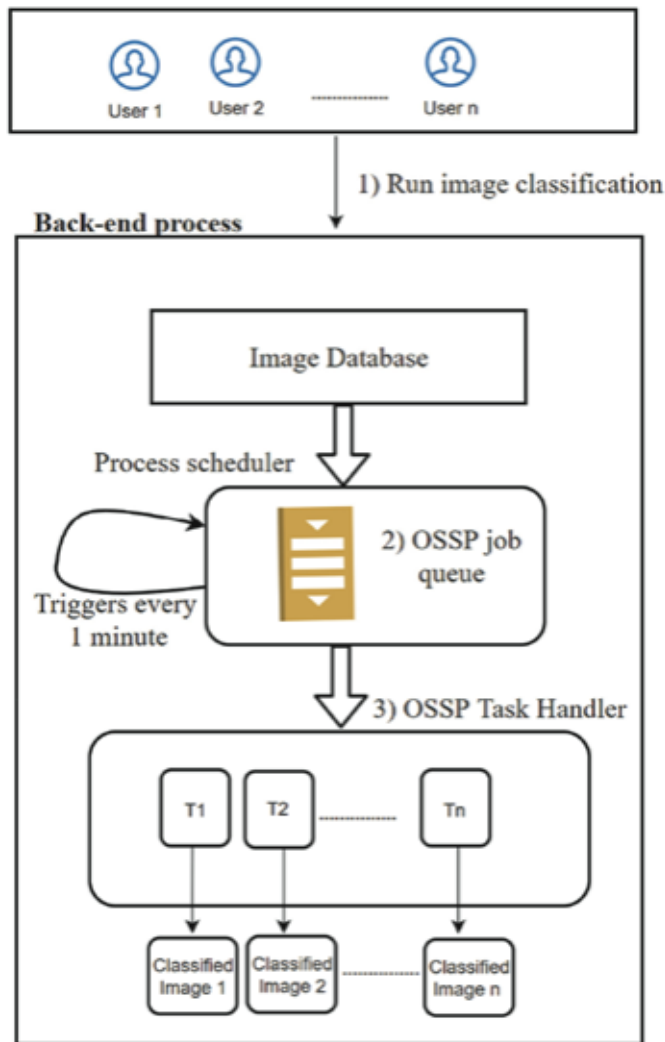


Figure 6. Image batch processing diagram demonstrates the sequence of actions for classification. OSSP—Open Source Sea-Ice Processing.

Clicking on the image marker displays a preview of the image along with its name.

4.2. Performance Comparison

To prepare the system for community adoption with good performance, we compared the performance of two types of experiments: (1) single-user batch processing with thread settings, and (2) multiple-user batch processing with different image input. The r5dn.24xlarge EC2 instance, with 768 GiB memory, was utilized for performance testing with a network bandwidth of 100 Gb/s, 96 logical processors, and 1 TB EBS volume. In Experiment 1, two mediums of batch processing of images are performed: (1) command prompt and (2) ArcCI platform. For each medium, a batch of 5, 10, and 20 images of 81.16 Mb, 146.8 Mb, and 328 Mb, respectively, were used. Each batch of images was processed at the following thread settings: 1, 2, 4, 8, 16, and 32.

4.2.1. Experiment 1. Single User Performs Batch Processing of Images Using the Command Prompt and ArcCI Platform

Figure 8A shows the processing time in the command prompt (CMD) and on the ArcCI platform. It is evident from the results that the time to classify images was reduced significantly in the ArcCI platform because it could classify multiple images in parallel, while the command prompt classified images one by one. Notably, in command prompt, performance decreased in the 16 and 32 threads because considerable time was consumed initializing the threads, and threads are underutilized. Figure 8B shows the maximum CPU utilization in the command prompt and in the ArcCI platform. Since the ArcCI platform could classify multiple images at a time, the CPU utilization was similar to that of the command prompt.

4.2.2. Experiment 2. Multiple Users Perform Batch Processing through ArcCI Platform

In Experiment 2, each user utilized 8 threads, while the number of users included 2, 4, 6, and 8. Each time, 10 images (146.8 Mb) were tested per user. Figure 9A shows the duration of processing for multi-users. The results show that the completion time increases as the number of users increases. As for the maximum CPU utilization percentage (Fig. 9B), there was no significant increase from 4 to 8 users. Notably, there is a direct correlation between the completion time and number of users, as well as between maximum CPU utilization and the amount of processing images.

3. CONCLUSIONS

This chapter described a cloud computing-based CI for collecting, organizing, searching, exploring, analyzing, visualizing, and sharing HSR images in the state-of-art AWS cloud environment using ML classification algorithms. This solution helped to address the challenges posed by the massive volume of HSR sea-ice aerial imagery, heterogeneous data sources, and the frequent update of new data. Additionally, the chapter introduced the implementation of a prototype of an online service for domain scientists to classify images and extract geophysical parameters. The ArcCI platform was developed to integrate existing time-series images. Specifically, the functionalities of the ArcCI web service include image data management, user management, batch image processing, results review, and spatiotemporal visualization modules.

To conclude, the ArcCI system was the first of its kind to support efficient storage of HSR images, on-demand services like batch image classification for single- or multi-user, and interactive spatiotemporal analysis of sea-ice evolution. To improve the Arctic CI laid out in this chapter, we identified four directions for future research. The first is to enhance the ArcCI system to autoscale dynamically. The second is to expand the scope of CI not just for polar science but to support research in other Earth science projects. The third is to include different categories of sea ice, such as new ice, young ice, first-year ice, etc., based on the

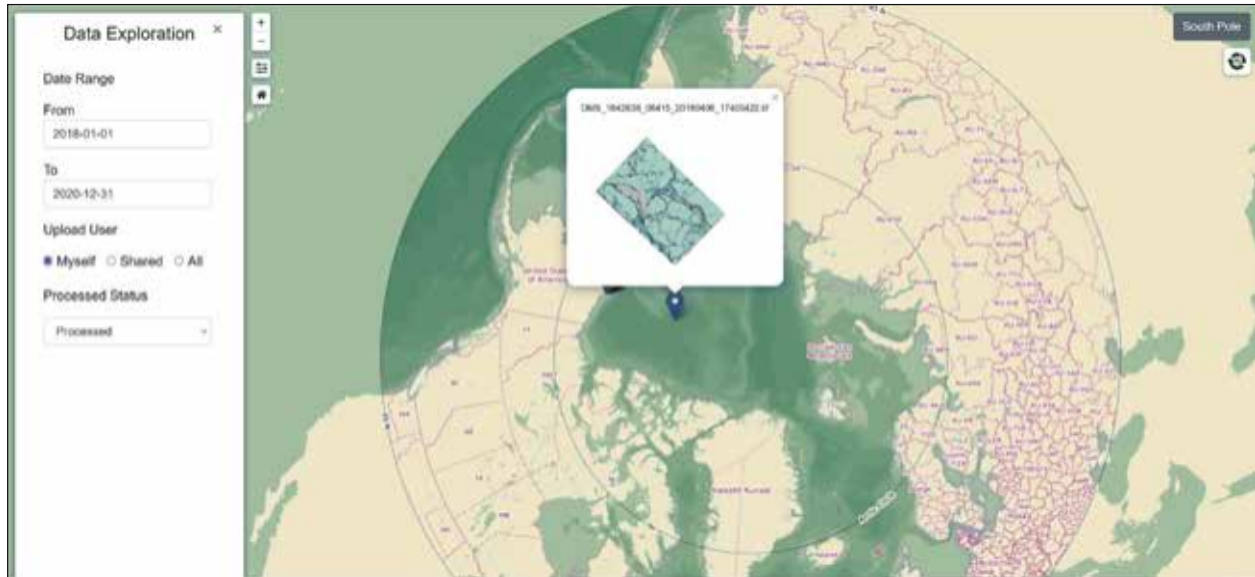


Figure 7. Arctic Data Exploration Tool is used to visualize the image location and classified image.

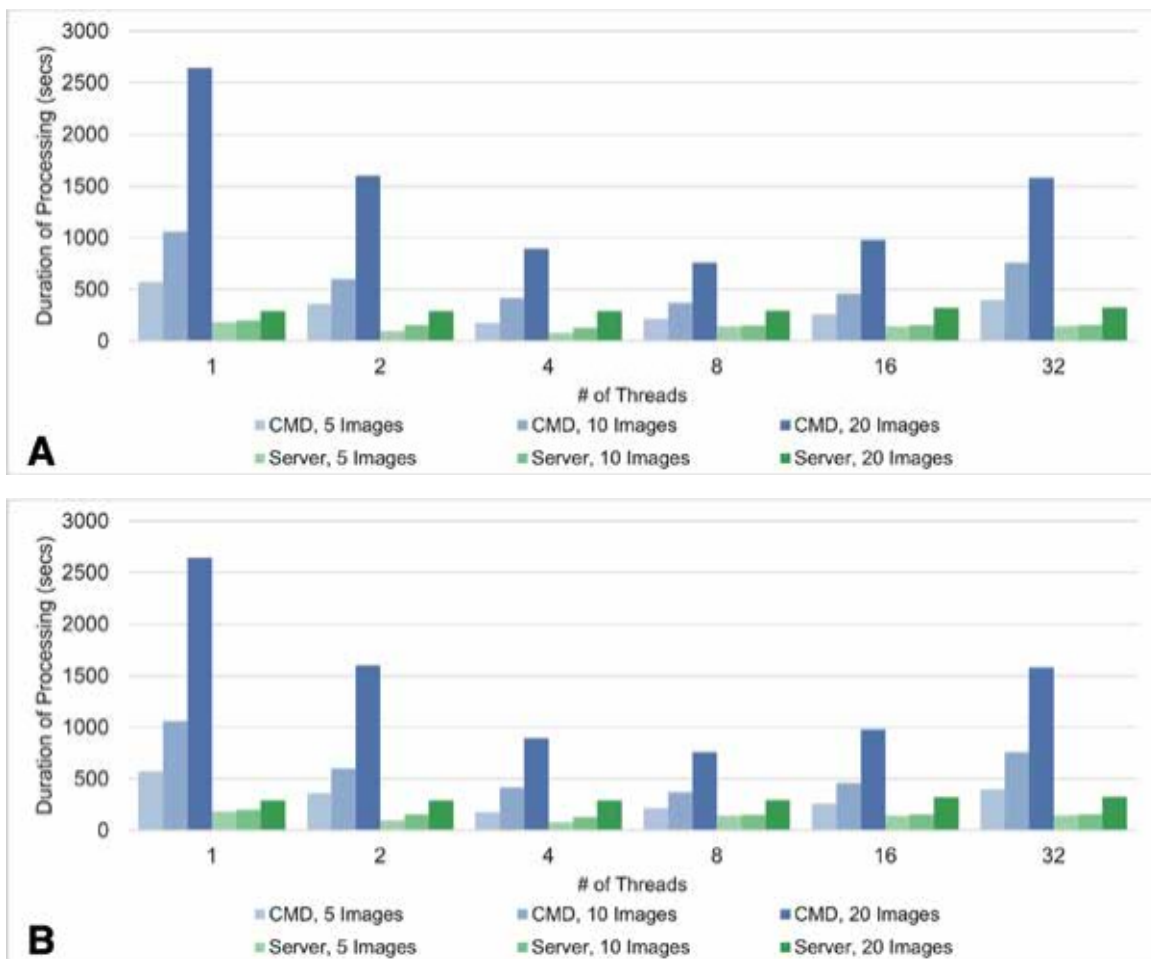


Figure 8. (A) Graph shows the duration of batch processing for a single user in command prompt (CMD) and ArcCI platform. (B) Maximum CPU utilization for single user in command prompt and ArcCI platform.

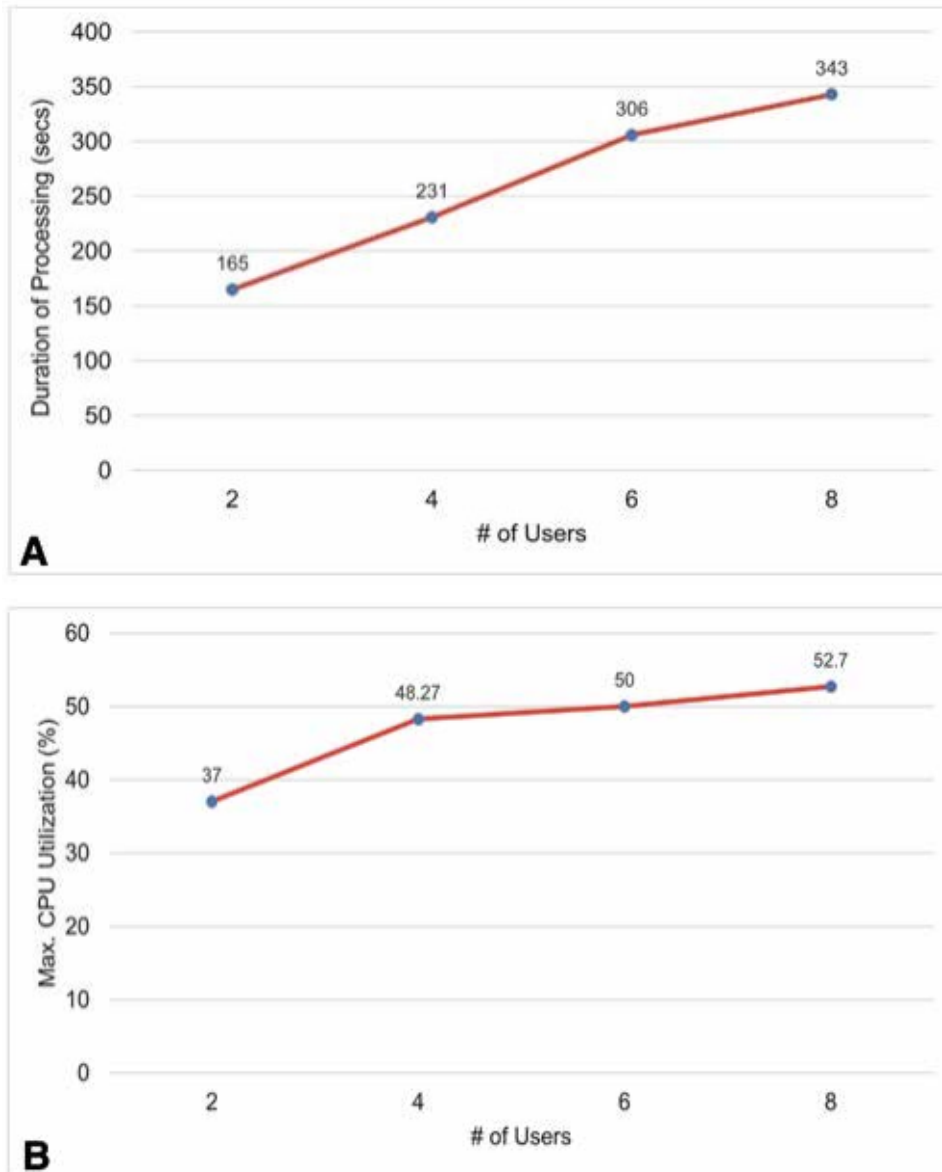


Figure 9. (A) Graph shows the batch processing duration for 2, 4, 6, and 8 users. (B) Maximum CPU utilization for 2, 4, 6, and 8 users.

World Meteorology Organization sea-ice nomenclature (Sea Ice Nomenclature, 2022). The fourth is to improve the sea-ice classification and detection accuracy using DL methodologies.

Available Open Access Resources: The code to build the CI and OSSP process are available at <https://github.com/stcenter/ArcCI>, and code to build the DL model is available at https://github.com/stcenter/ArcCI_DL. The ArcCI system URL is <https://arcciserver.stcenter.net/login.php>. The walk-through video to run the OSSP process is available at <https://youtu.be/VhIkhR-468Y>.

ACKNOWLEDGMENTS

The research presented in this chapter was funded by the National Science Foundation (1841520 and 1835507).

REFERENCES CITED

- Amazon, 2018, New C5n instances with 100 Gbps networking: <https://aws.amazon.com/blogs/aws/new-c5n-instances-with-100-gbps-networking/> (accessed January 2022).
- Amazon, 2022a, AWS Elastic Beanstalk: <https://aws.amazon.com/elasticbeanstalk/> (accessed January 2022).
- Amazon, 2022b, Amazon EC2: <https://aws.amazon.com/ec2/> (accessed January 2022).
- Amazon, 2022c, AWS Lambda: <https://aws.amazon.com/lambda/> (accessed January 2022).
- Amazon, 2022d, Amazon Relational Database Service (RDS): <https://aws.amazon.com/rds/> (accessed January 2022).
- Amazon, 2022e, Amazon S3: <https://aws.amazon.com/s3/?nc=sn&loc=1> (accessed January 2022).
- Amazon, 2022f, Amazon SageMaker: <https://aws.amazon.com/sagemaker/> (accessed January 2022).
- Amazon, 2022g, Amazon API Gateway: <https://aws.amazon.com/api-gateway/> (accessed January 2022).

- Ampatzidis, Y., Partel, V., and Costa, L., 2020, Agroviz: Cloud-based application to process, analyze and visualize UAV-collected data for precision agriculture applications utilizing artificial intelligence: *Computers and Electronics in Agriculture*, v. 174, <https://doi.org/10.1016/j.compag.2020.105457>.
- AWM, 2022, Arctic Web Map: https://webmap.arcticconnect.ca/#ac_3573/2/90.0/0.0 (accessed January 2022).
- Bellenger, D., Bertram, J., Budina, A., Koschel, A., Pfänder, B., Serowy, C., Astrova, I., Gatzu Grivas, S., and Schaaf, M., 2011, Scaling in cloud environments: *Recent Researches in Computer Science*, v. 33, p. 145–150.
- Biewald, L., 2020, Experiment tracking with weights and biases: Software available from <https://www.wandb.com/> (accessed October 2022).
- Bühler, Y., Adams, M.S., Bösch, R., and Stoffel, A., 2016, Mapping snow depth in alpine terrain with unmanned aerial systems (UASs): Potential and limitations: *The Cryosphere*, v. 10, p. 1075–1088, <https://doi.org/10.5194/tc-10-1075-2016>.
- C5n, Amazon, 2022, New C5n instances with 100 Gbps networking: <https://aws.amazon.com/blogs/aws/new-c5n-instances-with-100-gbps-networking/> (accessed January 2022).
- Chen, L.-C., Papandreou, G., Schroff, F., and Adam, H., 2017, Rethinking atrous convolution for semantic image segmentation: *CoRR abs/1706.05587*: <https://arxiv.org/abs/1706.05587> (accessed October 2022).
- Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., and Moore, R., 2017, Google Earth Engine: Planetary-scale geospatial analysis for everyone: *Remote Sensing of Environment*, v. 202, p. 18–27, <https://doi.org/10.1016/j.rse.2017.06.031>.
- Gui, Z., Yang, C., Xia, J., Li, J., Rezgüi, A., Sun, M., Xu, Y., and Fay, D., 2013a, A visualization-enhanced graphical user interface for geospatial resource discovery: *Annals of GIS*, v. 19, p. 109–121, <https://doi.org/10.1080/19475683.2013.782467>.
- Gui, Z., Yang, C., Xia, J., Liu, K., Xu, C., Li, J., and Lostritto, P., 2013b, A performance, semantic and service quality-enhanced distributed search engine for improving geospatial resource discovery: *International Journal of Geographical Information Science*, v. 27, p. 1109–1132, <https://doi.org/10.1080/13658816.2012.739692>.
- Huang, Y., Gao, P., Zhang, Y., and Zhang, J., 2018, A cloud computing solution for big imagery data analytics: 2018 International Workshop on Big Geospatial Data and Data Science (BGDDs): *IEEE*, p. 1–4, <https://doi.org/10.1109/BGDDs.2018.8626847>.
- Hussain, M., Chen, D., Cheng, A., Wei, H., and Stanley, D., 2013, Change detection from remotely sensed images: From pixel-based to object-based approaches: *International Journal of Photogrammetry and Remote Sensing*, v. 80, p. 91–106, <https://doi.org/10.1016/j.isprsjprs.2013.03.006>.
- Intel, 2022, What is hyper-threading?: <https://www.intel.com/content/www/us/en/gaming/resources/hyper-threading.html> (accessed January 2022).
- Jiang, Y., Li, Y., Yang, C., Liu, K., Armstrong, E., Huang, T., Moroni, D., and Finch, C., 2017, A comprehensive methodology for discovering semantic relationships among geospatial vocabularies using oceanographic data discovery as an example: *International Journal of Geographical Information Science*, v. 31, p. 2310–2328, <https://doi.org/10.1080/13658816.2017.1357819>.
- Jiang, Y., Li, Y., Yang, C., Hu, F., Armstrong, E.M., Huang, T., Moroni, D., McGibbney, L.J., Greguska, F., and Finch, C.J., 2018, A smart web-based geospatial data discovery system with oceanographic data as an example: *ISPRS International Journal of Geo-Information*, v. 7, <https://doi.org/10.3390/ijgi7020062>.
- Kulawiak, M., and Chybicki, A., 2018, Application of Web-GIS and geovisual analytics to monitoring of seabed evolution in South Baltic Sea coastal areas: *Marine Geodesy*, v. 41, p. 405–426, <https://doi.org/10.1080/01490419.2018.1469557>.
- Leaflet, 2022, Leaflet: <https://leafletjs.com/> (accessed January 2022).
- Li, Z., Yang, C.P., Wu, H., Li, W., and Miao, L., 2011, An optimized framework for seamlessly integrating OGC Web Services to support geospatial sciences: *International Journal of Geographical Information Science*, v. 25, p. 595–613, <https://doi.org/10.1080/13658816.2010.484811>.
- Li, Z., Yang, C., Jin, B., Yu, M., Liu, K., Sun, M., and Zhan, M., 2015, Enabling big geoscience data analytics with a cloud-based, MapReduce-enabled and Service-Oriented Workflow framework: *PLoS One*, v. 10, <https://doi.org/10.1371/journal.pone.0116781>.
- Li, Z., Yang, C., Huang, Q., Liu, K., Sun, M., and Xia, J., 2017, Building model as a service to support geosciences: *Computers, Environment and Urban Systems*, v. 61, p. 141–152, <https://doi.org/10.1016/j.compenvurbsys.2014.06.004>.
- Lightning, 2022, Pytorch Lightning: <https://www.pytorchlightning.ai/> (accessed January 2022).
- Liu, D., and Xia, F., 2010, Assessing object-based classification: Advantages and limitations: *Remote Sensing Letters*, v. 1, p. 187–194, <https://doi.org/10.1080/01431161003743173>.
- Lu, P., Li, Z., Cheng, B., Lei, R., and Zhang, R., 2010, Sea ice surface features in Arctic summer 2008: Aerial observations: *Remote Sensing of Environment*, v. 114, p. 693–699, <https://doi.org/10.1016/j.rse.2009.11.009>.
- Mattmann, C.A., 2013, A vision for data science: *Nature*, v. 493, p. 473–475, <https://doi.org/10.1038/493473a>.
- Nishar, A., Richards, S., Breen, D., Robertson, J., and Breen, B., 2016, Thermal infrared imaging of geothermal environments and by an unmanned aerial vehicle (UAV): A case study of the Wairakei–Tauhara geothermal field, Taupo, New Zealand: *Renewable Energy*, v. 86, p. 1256–1264, <https://doi.org/10.1016/j.renene.2015.09.042>.
- Open Geospatial Consortium, 2022a, Web Coverage Service: <https://www.ogc.org/standards/wcs> (accessed January 2022).
- Open Geospatial Consortium, 2022b, Web Feature Service: <https://www.ogc.org/standards/wfs> (accessed January 2022).
- Open Geospatial Consortium, 2022c, Web Map Service: <https://www.ogc.org/standards/wms> (accessed January 2022).
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S., 2019, PyTorch: An imperative style, high-performance Deep Learning Library: *CoRR abs/1912.01703*, <https://arxiv.org/abs/1912.01703>.
- PHP, 2022, What is PHP?: <https://www.php.net/manual/en/intro-what-is.php>: <https://www.ogc.org/standards/wcs> (accessed January 2022).
- PostgreSQL, 2022, What is PostgreSQL?: <https://www.postgresql.org/about/> (accessed January 2022).
- RCLONE, 2022, RClone Mount: https://rclone.org/commands/rclone_mount/ (accessed January 2022).
- Renner, A.H.H., Dumont, M., Beckers, J., Gerland, S., and Haas, C., 2013, Improved characterisation of sea ice using simultaneous aerial photography and sea ice thickness measurements: *Cold Regions Science and Technology*, v. 92, p. 37–47, <https://doi.org/10.1016/j.coldregions.2013.03.009>.
- Sawant, S., and Mohite, J., 2018, Towards internet of things based approach for using archives of Earth observation for crop water management in semi-arid areas: *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, p. 3437–3440, <https://doi.org/10.1109/IGARSS.2018.8517521>.
- Seier, G., Kellerer-Pirklbauer, A., Wecht, M., Hirschmann, S., Kaufmann, V., Lieb, G.K., and Sulzer, W., 2017, UAS-based change detection of the glacial and proglacial transition zone at Pasterze Glacier, Austria: *Remote Sensing*, v. 9, <https://doi.org/10.3390/rs9060549>.
- Sha, D., 2021, Geophysical feature extraction and spatiotemporal analysis of polar sea ice using high spatial resolution imagery [Ph.D. dissertation]: Fairfax, Virginia, George Mason University, 84 p.
- Sha, D., Miao, X., Xu, M., Yang, C., Hongjie Xie, H., Mestas-Nuñez, A.M., Li, Y., Liu, Q., and Yang, J., 2020, An on-demand service for managing and analyzing Arctic sea ice high spatial resolution imagery: *Data*, v. 5, <https://doi.org/10.3390/data5020039>.
- Shapiro, L.G., and Stockman, G.C., 2001, *Computer Vision: Upper Saddle River, New Jersey*, Prentice Hall, 608 p.
- Tamiminia, H., Salehi, B., Mahdianpari, M., Quackenbush, L., Adeli, S., and Brisco, B., 2020, Google Earth Engine for geo-big data applications: A meta-analysis and systematic review: *ISPRS Journal of Photogrammetry and Remote Sensing*, v. 164, p. 152–170, <https://doi.org/10.1016/j.isprsjprs.2020.04.001>.
- Torchmetrics, 2022, Torchmetrics: <https://torchmetrics.readthedocs.io/en/latest/> (accessed January 2022).
- Walker Johnson, G., Gaylord, A.G., Franco, J.C., Cody, R.P., Brady, J.J., Manley, W., Dover, M., Garcia-Lavigne, D., Score, R., and Tweedie, C.E., 2011, Development of the Arctic Research Mapping Application (ARMAP): Interoperability challenges and solutions: *Computers & Geosciences*, v. 37, p. 1735–1742, <https://doi.org/10.1016/j.cageo.2011.04.004>.
- WampServer, 2022, WampServer: <https://www.wampserver.com/en/> (accessed January 2022).
- Wang, Q., Danilov, S., Jung, T., Kaleschke, L., and Wernecke, A., 2016, Sea ice leads in the Arctic Ocean: Model assessment, interannual variability and trends: *Geophysical Research Letters*, v. 43, p. 7019–7027, <https://doi.org/10.1002/2016GL068696>.

- WMO Sea ice nomenclature, 2022: https://library.wmo.int/doc_num.php?explnum_id=4651 (accessed April 2022), 121 p.
- Wright, N.C., and Polashenski, C.M., 2018, Open-source algorithm for detecting sea ice surface features in high-resolution optical imagery: *The Cryosphere*, v. 12, p. 1307–1329, <https://doi.org/10.5194/tc-12-1307-2018>.
- Xie, H., Tian, Y.Q., Granillo, J.A., and Keller, G.R., 2007, Suitable remote sensing method and data for mapping and measuring active crop fields: *International Journal of Remote Sensing*, v. 28, p. 395–411, <https://doi.org/10.1080/01431160600702673>.
- Yang, C., Raskin, R., Goodchild, M., and Gahegan, M., 2010, Geospatial cyberinfrastructure: Past, present and future: *Computers, Environment and Urban Systems*, v. 34, p. 264–277, <https://doi.org/10.1016/j.compenvurbsys.2010.04.001>.
- Yang, C., Goodchild, M., Huang, Q., Nebert, D., Raskin, R., Xu, Y., Bambacus, M., and Fay, D., 2011, Spatial cloud computing: How can the geospatial sciences use and help shape cloud computing?: *International Journal of Digital Earth*, v. 4, p. 305–329, <https://doi.org/10.1080/17538947.2011.587547>.
- Yang, C., Huang, Q., Li, Z., Xu, C., and Liu, K., 2013, *Spatial Cloud Computing: A Practical Approach*: Boca Raton, Florida, CRC Press, 357 p., <https://doi.org/10.1201/b16106>.
- Yelick, K., Coghlan, S., Draney, B., Ramakrishnan, L., Scovel, A., Sakrejda, I., et al., 2011, *The Magellan Report on Cloud Computing for Science*, U.S. Department of Energy, Office of Advanced Scientific Computing Research, 138 p. + appendices, <https://doi.org/10.2172/1076794>.
- Yuan, X., Shi, J., and Gu, L., 2021, A review of deep learning methods for semantic segmentation of remote sensing imagery: *Expert Systems with Applications*, v. 169, <https://doi.org/10.1016/j.eswa.2020.114417>.
- Zheng, P., Wu, Z., Zhang, W., Li, M., Yang, J., Zhang, Y., and Wei, Z., 2018, An unmixing-based content retrieval method for hyperspectral imagery repository on cloud computing platform: *IGARSS 2018–2018 IEEE International Geoscience and Remote Sensing Symposium*, p. 3583–3586, <https://doi.org/10.1109/IGARSS.2018.8517591>.
- Zhuang, J., Jacob, D.J., Gaya, J.F., Yantosca, R.M., Lundgren, E.W., Sulprizio, M.P., and Eastham, S.D., 2019, Enabling immediate access to Earth science models through cloud computing: Application to the GEOS-Chem model: *Bulletin of the American Meteorological Society*, v. 100, p. 1943–1960, <https://doi.org/10.1175/BAMS-D-18-0243.1>.
- Zhuang, J., Jacob, D.J., Lin, H., Lundgren, E.W., Yantosca, R.M., Gaya, J.F., Sulprizio, M.P., and Eastham, S.D., 2020, Enabling high-performance cloud computing for Earth science modeling on over a thousand cores: Application to the GEOS-Chem Atmospheric Chemistry Model: *Journal of Advances in Modeling Earth Systems*, v. 12, <https://doi.org/10.1029/2020MS002064>.

Manuscript Accepted by the Society 20 April 2022
 Manuscript Published Online 8 February 2023