Improving Satellite Imagery Masking Using Multitask and Transfer Learning

Rangel Daroya, Luisa Vieira Lucchese, Travis Simmons, Punwath Prum, Tamlin Pavelsky, John Gardner, Colin J. Gleason, and Subhransu Maji

***Abstract*—Many remote sensing applications require masking of pixels in satellite imagery for further analysis. For instance, estimating water quality variables such as suspended sediment concentration (SSC) requires isolating pixels depicting water bodies unaffected by clouds, their shadows, terrain shadows, and snow and ice formation. A significant bottleneck is the reliance on multiple data products (e.g., satellite imagery and elevation maps) and lack of precision in individual processing steps, which degrade estimation accuracy. We propose a unified masking system that predicts all necessary masks from harmonized landsat and sentinel (HLS) imagery. Our model leverages multitask learning to improve accuracy while sharing computation across tasks for added efficiency. In this article, we explore recent deep learning architectures, demonstrating that masking performance benefits from pretraining on large satellite imagery datasets. We present a range of models offering different speed/accuracy tradeoffs: MobileNet variants provide the fastest inference while maintaining competitive accuracy, whereas transformer-based architectures achieve the highest accuracy, particularly when pretrained on large-scale satellite datasets. Our models provide a 9% $F1$ score improvement compared to previous work on water pixel identification. When integrated with an SSC estimation system, our models result in a 30× speedup while reducing estimation error by 2.64 mg/L, allowing for global-scale analysis. We also evaluate our model on a recently proposed cloud and cloud shadow estimation benchmark, where we outperform the current state-of-the-art model by at least 6% in $F1$ score.**

***Index Terms*—Deep learning, global surface water detection, multitask learning, suspended sediment, transfer learning.**

I. Introduction

ISOLATING different types of pixels is a prerequisite in many remote sensing tasks. Prior to estimating a variable of interest, it is often crucial to isolate pixels covering specific land cover types such as water [1], [2], forests [3], [4], agriculture [5], [6], urban areas [7], [8], or remove artifacts such as clouds and shadows [3], [9]. Landsat and Sentinel-2 data come with water masks processed with the function of mask (Fmask) algorithm [10], [11], but these are often insufficient due to the tendency of Fmask to miss clouds [9], misclassify water in complex environments [12], and misclassify clear ice as open water [13]. Therefore, remote sensing workflows across many domains have developed specific preprocessing and masking tasks essential in estimating a quantity of interest [9], [14], [15].

In this article, we demonstrate a new approach to masking for a case of estimating suspended sediment concentration (SSC). Although SSC is discussed here as an application, our proposed masking procedure is broadly applicable to other remote sensing estimation methods. SSC can be estimated in rivers from a given satellite image by first classifying the pixels into water and nonwater classes to identify valid areas for analysis [1], [15], [16]. All artifacts from clouds, cloud shadows, terrain shadows, snow, and ice, considered as nonwater, are also typically removed so that only “good quality” water pixels remain [15]. Once good quality pixels are identified, summary statistics such as mean, variance, and counts from the reflective bands of water pixels are used as input to train a model to predict SSC [17], [18], [19]. Threshold-based classification of pixels such as via the Fmask algorithm [10] and the modified normalized difference water index (MNDWI) [20] can be effective for water identification, but they are highly dependent on the locations and weather conditions where thresholds are determined for the different types of masks [21]. Fmask applies spectral tests to thermal and optical bands for identifying cloud, shadow, water, and snow pixels with thresholds based on the global optima across sampled reference images [10]. However, a lower threshold for identifying clouds that works well in areas with predominantly thin clouds could result in overestimation of clouds in areas with no clouds at all. Similar issues arise with other masks, where the distribution of dark surfaces (e.g., burned areas, wetlands) could affect the prediction of shadows and water due to similar spectral characteristics. MNDWI thresholds the ratio involving the green and short-wavelength infrared (SWIR) bands since radiation from SWIR is strongly absorbed by water [20]. However, it is also sensitive to shadows from different topographic conditions [22] and is very sensitive to snow; it is identical in form to the normalized difference snow index [23].

Received 7 September 2024; revised 9 December 2024; accepted 1 March 2025. Date of publication 14 March 2025; date of current version 4 April 2025. This work was supported in part by the NASA AIST program under Grant 80NSSC22K1487 and in part by the National Science Foundation under Grant 2329927 and Grant 1749833. *(Corresponding author: Rangel Daroya.)*

Rangel Daroya and Subhransu Maji are with the College of Information and Computer Sciences, University of Massachusetts, Amherst, MA 01003 USA (e-mail: rdaroya@umass.edu; smaji@cs.umass.edu).

Luisa Vieira Lucchese, Punwath Prum, and John Gardner are with the Department of Geology and Environmental Science, University of Pittsburgh, Pittsburgh, PA 15260 USA (e-mail: luisa.lucchese@pitt.edu; punwath.prum@pitt.edu; gardner.john@pitt.edu).

Travis Simmons and Colin J. Gleason are with the Department of Civil and Environmental Engineering, University of Massachusetts, Amherst, MA 01003 USA (e-mail: tsimmons@umass.edu; cjgleason@umass.edu).

Tamlin Pavelsky is with the Department of Earth, Marine and Environmental Science, University of North Carolina, Chapel Hill NC 27599 USA (e-mail: pavelsky@unc.edu).

Digital Object Identifier 10.1109/JSTARS.2025.3551620

Once water is obtained (Fmask also returns snow, ice, and clouds following the same threshold-based logic), water pixels must be sorted into those of "good" quality. The definition of "good" is subjective, but often sun glint and shadows are problematic for reflectance based applications [24], [25]. Ancillary data are needed to model terrain shadows. Shadows can be obtained using the approximate position of the sun (based on the date and time of day) and the surrounding topography [using digital elevation maps (DEM)]. DEMs are large files, especially those that are global [26], so this shadowing process is data intensive. While we have given examples of preprocessing for SSC estimation, the same issues apply for other applications that require the isolation of pixels to use the surface reflectance to drive another estimation process.

Deep learning networks provide an alternative to threshold-based methods; previous work has shown the promising performance of deep learning models for segmenting images where each pixel is classified to a specific class [27], [28], [29], [30]. In particular, semantic segmentation has been explored together with deep learning methods similar to [30] where a fully convolutional neural network (CNN) is used to detect cloud and cloud shadows. DeepWaterMap (DWM) [29] uses a CNN to segment water pixels from Landsat data. LANA [9] uses an improved CNN with attention-based mechanisms [31] to identify cloud and shadow pixels from Landsat 8 data. Other architectures, such as DeepLabv3+ [32], MobileNet [33], SegNet [34], vision transformers (ViT) [35], and swin transformers (Swin-T) [36] have shown encouraging results for object-based pixel classification in images.

Apart from architecture considerations, deep learning methods require a sufficient amount of labeled training data to achieve competitive performance [37], [38], [39]. To reduce dependency on large datasets, transfer learning is applied where models are first trained on a different task (e.g., image classification) with a large dataset (e.g., ImageNet [39] with more than 14 M images) and then fine-tuned on a specific task (e.g., masking) [40], [41], [42]. Several large datasets such as ImageNet [39], Prithvi [43], and Satlas [44] have become available with millions of labeled training images that were shown to be effective for pretraining. At the same time, fine-tuning can be applied using the recently released dynamic surface water extent (DSWx) [45] product associated with the harmonized landsat-sentinel (HLS) project [46] contains labeled masking data with high frequency and near-global coverage. It has been used for flood detection, wildfire mapping, and reservoir monitoring using changes in water bodies and vegetation, and can potentially be used for masking applications (e.g., water, cloud, shadow identification) [47]. While DSWx can be used as the sole source of masks, it is limited by the required input sources to produce them (Copernicus DEM, Copernicus land cover, ESA worldcover, NOAA GSHHS shapefile, and HLS). DSWx is currently not available for historical data and it would be challenging and resource-intensive to generate [45]. In contrast, deep learning models can predict masks without relying on multiple sources (e.g., predict masks using only HLS).

With the increasing availability of datasets for both pretraining and task-specific fine-tuning, deep learning models show great potential to find patterns and generalize on unseen data for satellite imagery masking. However, these architectures typically use a single model to predict a single output. In the case of predicting five masks (i.e., water, cloud, cloud shadow, snow/ice, terrain shadow) as is required for SSC estimation, five such models would be necessary, which would require more time and resources for training and inference. Multitask models [48] introduce a framework that allows the simultaneous prediction of multiple outputs (e.g., water, shadow, cloud masks). It was shown to improve generalization performance by learning multiple tasks at the same time, while simplifying the training to a single model [49]. Instead of training five separate models where each model predicts a single type of mask (e.g., water), a multitask framework uses a single model to predict all five masks (water, cloud, cloud shadow, terrain shadow, snow/ice). Thus, approximately five times less resources would be needed for training and inference.

In this article, we propose an end-to-end framework for estimating SSC using a multitask deep learning model. We investigate the reliability and efficiency of identifying water pixels and other artifacts (e.g., clouds, cloud shadows, terrain shadows, snow/ice) from satellite imagery using multitask models and transfer learning through different pretraining datasets (e.g., ImageNet, Satlas, Prithvi), and compare our results with existing models such as LANA [9] and DWM [50]. We also compare performance of combining single-task models against the multitask equivalent. Finally, we show the impact of multitasking by evaluating the effect of masking improvements on downstream tasks—here using SSC estimation as a case study. Our code is available at https://github.com/cvl-umass/improv-mask.

II. Data Sources

A. Harmonized Landsat-Sentinel

The HLS project [46] uses the operational land imager (OLI) and multispectral instrument sensors from the Landsat and Sentinel remote sensing satellites. The combined temporal frequency is 2–3 days at 30 m spatial resolution with Landsat data starting from February 2013, and Sentinel-2 A/2B starting from June 2015/March 2017. The satellites cover all land areas in the globe except Antarctica. The HLS data provide 15 harmonized bands as enumerated in Table I.

Input features for training the model were derived from HLS—the bands Blue, Green, Red, near infrared (NIR), SWIR1, SWIR2 were used, similar to DWM [29].

B. Annotated Global Surface Water Masks

More than 3 million tiles are available in the DSWx products through NASA's Earth Observing System Data and Information System (EOSDIS) [45]. DSWx products cover inputs generated from Sentinel-1, NISAR, and HLS. This work focuses primarily on the DSWx product that uses HLS as the image-based input. The product provides a map of the extent of surface waters across all landmasses excluding Antarctica. Each tile has ten layers, where each layer has a size of 3660 × 3660 pixels (each pixel has a 30 m resolution).

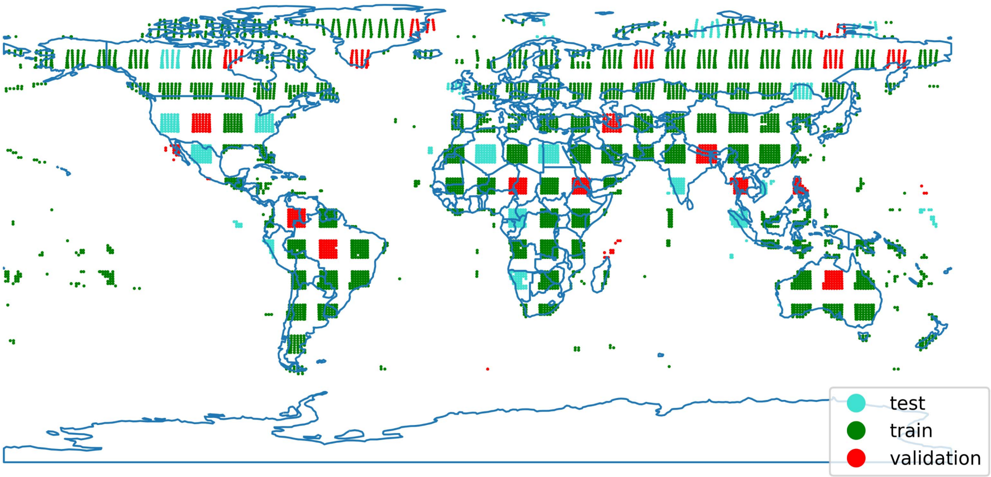

Fig. 1. *Geographic distribution of train, validation, and test data.* The dataset has a global coverage with train, validation, and test splits spanning different locations. Each dot represents the center of the sampled tile. Gaps are present to prevent overlap between different data splits, since each tile has a coverage of 109.8 km × 109.8 km. Following the computer science convention, training set is used for updating model weights during training, validation set is used for selecting hyperparameters, and test set is not seen by the model except when evaluating performance.

TABLE I
15 Harmonized Bands Present in HLS

Band	Wavelength
Coastal Aerosol	$0.32 - 0.45\mu$m
Blue	$0.45 - 0.51\mu$m
Green	$0.53 - 0.59\mu$m
Red	$0.64 - 0.67\mu$m
Red-edge 1	$0.69 - 0.71\mu$m
Red-edge 2	$0.73 - 0.75\mu$m
Red-edge 3	$0.77 - 0.79\mu$m
NIR broad	$0.78 - 0.88\mu$m
NIR narrow	$0.85 - 0.88\mu$m
Short-Wave Infrared 1 (SWIR-1)	$1.57 - 1.65\mu$m
SWIR-2	$2.11 - 2.29\mu$m
Water Vapor	$0.93 - 0.95\mu$m
Cirrus	$1.36 - 1.38\mu$m
Thermal Infrared 1	$10.60 - 11.19\mu$m
Thermal Infrared 2	$11.50 - 12.51\mu$m

Labels were obtained from a year's worth of DSWx data from April 2023 to March 2024 to train a model to predict the different masks. Labels included the five masks for water, cloud, cloud shadow, snow/ice, and terrain shadow (from DEM [26]). Only high confidence water pixels (both partial and open water) were used as water mask labels based on the DSWx product. The time frame was selected to obtain samples from varying seasons and weather conditions. Each label from the DSWx tiles were matched to available HLS tiles, and sampled as follows.

1) *Temporally*: for each tile, 20 data points were uniformly sampled throughout the year to cover different seasons and weather conditions. This is roughly 1 sample every 2 weeks.
2) *Spatially*: The globe was subdivided onto a 6° × 6° grid. All available tiles contained in each cell were used. This process was done to make sure that adjacent cells would have no overlapping tiles.

In addition, satellite tiles that did not contain data (i.e., all pixels flagged as having no data) were removed from the dataset. Samples with incomplete labels were also removed such that samples that had all 5 masks remained. Each HLS feature paired with a DSWx label was cropped to a size of 512 × 512 pixels. We applied spatial validation on the data to prevent data leakage and to measure the predictive performance of the model on unseen locations. Train, validation, and test data points were based on the grid defined during sampling. Fig. 1 shows the distribution of the sampled data. The extraction and processing resulted in 107 250 labeled data—82 247 for training, 12 849 for validation, and 12 154 for testing. Throughout this article, we use the computer science convention for defining train, validation, and test splits. The training samples are used for training the model, the validation samples are used for selecting hyperparameters and thresholds (if any), and the testing samples are exclusively used to evaluate the performance of the models. Table II shows the average number of pixels per image sample across the different data splits. Different classes across training, validation, and test have a similar number of pixels per mask.

C. In Situ SSCs

SSCs were obtained from water quality databases [15], [51] across different locations around the globe. Data were collected in situ in different sites in the United States (U.S. Geological Survey, 2018), Canada (The Water Survey of Canada, 2018), South America (Agência Nacional de Águas, 2017), Taiwan (Taiwan Water Resource Agency, 2018), and Europe (European Environment Agency, 2020). Additional databases such as GEMStat [52] and Glorich [53] also have metadata that indicate the quality of the measurements and the depth at which the concentrations were measured.

The SSC data were obtained from the United States, Canada, South America, Europe, and parts of Asia (e.g., Taiwan, Japan,

TABLE II
AVERAGE NUMBER OF PIXELS PER IMAGE PER TYPE OF MASK ACROSS ALL SAMPLES IN THE DATA SPLITS

Data split	Water	Cloud shadow	Cloud	Snow/ice	Terrain shadow
Train	51,042	43,709	229,906	32,318	545,703
Validation	42,142	40,588	210,917	39,278	547,651
Test	36,213	40,916	213,802	22,397	552,310

China). A total of 244 000 in situ SSC data were obtained. HLS tiles and in situ SSC values were then matched based on the location and date. Satellite images taken within one day of the sampled SSC value, and images with nonzero data were used. Using this matching criteria, we obtained 24 328 data points with both in situ SSC values and corresponding HLS tiles. These data points were split into train, validation, and test with 50%, 25%, and 25% of the data, respectively. These three sets were sampled to be in different spatial locations to prevent data leakage, and to ensure an accurate assessment of the model's generalizability. The final set of SSC values range from 0.003 to 723.0 mg/L, taken from April 2013 to October 2021 [54].

We removed samples that could not be reliably measured. The metadata of each SSC sample includes the minimum measurable amount at that location, which is defined by the specific sensor used for SSC measurement. We define reliable samples as those that are above the given minimum measurable amount. We additionally only use samples near the river surface taken at a depth of at most one meter, since the penetration of light—used for obtaining image data—is at most only a few meters (less for turbid water).

III. METHODS

A. Multitask Model

Fig. 2 shows how single-task models differ from multitask models. When predicting five masks, single-task models would require five separate models, each trained and run separately. At the same time, five models are stored, requiring approximately five times as much storage compared to having a single model that predicts all masks at the same time. The multitask model trains on five different masks with one single backbone model that extracts a general feature. From a general feature, five small "heads" for each of the masks is trained that is composed of a few trainable layers (much smaller than the backbone). Instead of five large models trained separately, the multitask model only has to train one large model and five small sets of trainable layers simultaneously. The multitask framework would, thus, save time and resources for training and inference.

More formally, a single model is a parameterized function f that transforms an input x into output $\hat{y}$ by learning the function parameters θ : $f_\theta(x) = \hat{y}$. The parameters θ are learned by training f on a dataset of n image-label pairs using a specified loss function. Given a training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ where x_i is the input image and y_i is the corresponding label, the goal is to have the model output $f_\theta(x_i) = \hat{y}_i$ be as close as possible to y_i. In our work, $x_i \in \mathbb{R}^{512\times512\times6}$ is the 512×512 pixel image composed of six bands from HLS (red, green, blue, NIR, SWIR-1, SWIR-2), and $y_i \in \mathbb{R}^{512\times512\times1}$ is one of water, cloud, cloud shadow, snow/ice, or terrain shadow mask. For a multitask model that outputs five masks at the same time, there would be five labels y_i^m and five corresponding model outputs $\hat{y}_i^m$ where $m \in$ [water, cloud, cloud shadow, snow/ice, terrain shadow]. The multitask model can then be represented as follows:

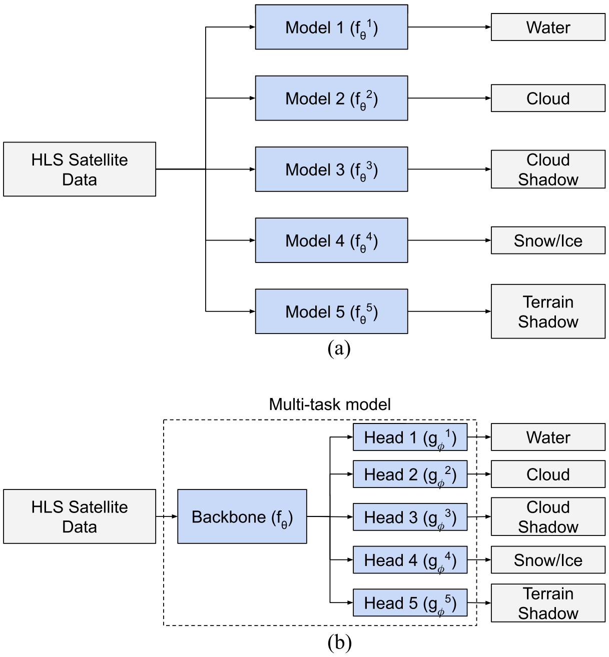

Fig. 2. *Comparison of multiple single-task models and multitask model.* (a) Evaluating a model for each output is resource intensive, since it would require running five separate models. (b) Multitask model setup where only one model is predicting all five outputs at the same time, using approximately one-fifths of the resources for training.

$$z_i = f_\theta(x_i) \tag{1}$$

$$y_i^m = g_{\phi_m}^m(z_i). \tag{2}$$

Equation (1) computes the general feature used for all masks in the model. To compute each mask m, there are five smaller networks g^m parameterized by ϕ_m—the networks have the same structure but have different parameters for each mask m—so that each output mask can be obtained through (2). Both f and g are trained simultaneously using the loss function in (3) where $\mathcal{L}_{\text{bce}}$ is the binary cross entropy loss computed for each of the five labels. $\mathcal{L}$ is minimized by optimizing the parameters θ and ϕ_m for all m

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\sum_{m}\mathcal{L}_{\text{bce}}\left(\hat{y}_i^m, y_i^m\right). \tag{3}$$

The binary cross entropy loss $\mathcal{L}_{\text{bce}} \in [0, \infty)$ can be computed by comparing all pixels in the prediction $\hat{y}_i^m$ and the ground truth y_i^m. For an image with $W \times H$ pixels, (4) shows the loss between the predicted and ground truth for mask m. The variable $y_{i,(j,k)}^m \in \{0, 1\}$ is the label or ground truth value for the pixel in position (j, k), and $\hat{y}_{i,(j,k)}^m \in [0, 1]$ is the predicted probability of the mask for the pixel in position (j, k)

$$\mathcal{L}_{\text{bce}}\left(\hat{y}_i^m, y_i^m\right) = \frac{1}{WH} \sum_{j=1}^{W} \sum_{k=1}^{H} -\Bigg(y_{i,(j,k)}^m \log \hat{y}_{i,(j,k)}^m + \left(1 - y_{i,(j,k)}^m\right) \log\left(1 - \hat{y}_{i,(j,k)}^m\right)\Bigg). \quad (4)$$

Using a multitask model, we explore different architectures and setups for the backbone, which is crucial for extracting useful features to predict the different masks simultaneously.

1) Network Architectures: CNNs [55] and transformers [35] are two main types of model architectures that have recently emerged for various computer vision tasks such as masking/segmentation [34], [56], image classification [35], [57], and object detection [40], [58]. While both architectures are effective, they have different compositions. CNNs use a learnable kernel that slides across an image to extract features [55]; transformers [35], [36] decompose an image into patches and use attention [31] to learn global dependencies and better context across the whole image. We describe both architectures in more detail as follows.

Convolutional neural networks: CNNs are networks where the model f encodes translational invariance and spatial locality. Translational invariance and spatial locality are achieved through the convolution layer, a key operator in CNNs. The convolution layer learns a kernel $K \in \mathbb{R}^{k\times k}$ applied to an image $I \in \mathbb{R}^{W\times H}$. Equation (5) shows the operation where S_{ij} is the output of the layer at the input image pixel location (i, j). The same operation is applied to all pixel positions (i, j) in I to form a feature map $S \in \mathbb{R}^{(W-k+1)\times(H-k+1)}$. Other layer parameters such as stride, dilation, and padding can also control the size of the feature map

$$S_{ij} = (I * K)_{ij} = \sum_{a=0}^{k-1} \sum_{b=0}^{k-1} I_{i+a,j+b} K_{a,b} \quad \forall i, j. \quad (5)$$

In addition to convolution layers, other operators such as pooling layers, nonlinear activations, and fully connected layers are also used to create CNNs. Pooling layers contribute to the translational invariance of CNNs by dividing the input image into patches, and aggregating each patch (e.g., by taking the maximum) to produce smaller feature maps. Nonlinear activations apply nonlinear transformations to the input to learn complex relations between inputs and outputs. Some examples include rectified linear unit: $\text{ReLU}(x) = \max(0, x)$ [59] and sigmoid: $\sigma(x) = 1/(1 + \exp(-x))$ [60]. Fully connected layers apply a linear transformation to all pixels in the input to produce a value in the output. Multiple linear transformations could be applied simultaneously to all input pixels to output multiple values.

CNNs are created by stacking a series of layers described above. Typical sequences are composed of multiple blocks of convolution—pooling—nonlinear. DeepLabv3+ [32], MobileNet [33], SegNet [34], U-Net [61], and ResNet [62] are commonly used architectures for various vision applications (e.g., object detection, segmentation, classification) that use a combination of these layers.

Transformers: Transformers do not have inductive biases on spatial locality that are present in CNNs due to the latter's use of sliding window kernels. As a result, transformers can learn global features that could perform better than CNNs [35]. The attention layer [63] is a critical block in transformers that models long-range dependencies in images to learn a global representation. To apply it, the input image is divided into n patches and flattened into a d-dimensional vector to have an input $X \in \mathbb{R}^{n\times d}$. Each attention layer learns a set of query ($W^Q \in \mathbb{R}^{d\times d_q}$), key ($W^K \in \mathbb{R}^{d\times d_k}$), and value ($W^V \in \mathbb{R}^{d\times d_v}$) weight matrices. These are applied to the input to produce queries $Q = XW^Q$, keys $K = XW^K$, and values $V = XQ^V$. The output of the attention layers is then computed [see (6)]. Additional learnable parameters such as positional embeddings are also added to the input patches to give information on the original position of the image patch [35]

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V. \quad (6)$$

Similar to CNNs, a transformer model f stacks a series of blocks and use other layers such as nonlinear activations and fully connected layers. ViT [35] and Swin-T [36] are commonly used architectures applied to vision tasks such as classification and segmentation.

2) Transfer Learning: Deep learning models have millions of parameters that need to be optimized during training. When only a small dataset is available for training, there is a risk of model overfitting, where the model simply memorizes the training data and does not learn to generalize to unseen samples. To mitigate this risk, transfer learning first trains the model on a very large dataset and then fine-tunes it on a separate dataset for the specific application [40], [41], [42]. Training on the larger dataset teaches the model to extract useful feature maps from images that could be helpful for a general understanding of images, which could then be used for applications such as classification [40].

Building on this idea, we apply transfer learning by first pretraining the backbone model f_θ [see Fig. 2(b)] on a different but much larger dataset to learn the parameters $\hat{\theta}$. Then, we replace the last layer with the task-specific heads (g_ϕ) and train the model in Fig. 2(b) using the annotated global surface masks (introduced in Section II-B) and with the backbone parameters θ initially set to $\hat{\theta}$. Unlike other methods that start the optimization from random parameters, we optimize the multitask model starting from parameters that can already extract acceptable features for understanding images (i.e., we apply transfer learning from other larger existing datasets).

ImageNet [39], Satlas [44], and Prithvi [43] were explored for pretraining in this work. ImageNet is an image classification dataset of 1000 object classes containing more than 1 million training images, 50 000 validation images, and 100 000 test images. Pretraining on ImageNet has shown impressive results on image classification [64], image segmentation [65], and object detection [66]. Satlas was recently introduced as a remote sensing dataset that combines images from Sentinel-2 and National Agriculture Imagery Program to produce 302 million labeled images for various tasks including classification, regression, object detection, and segmentation. Their work shows competitive performance against other ImageNet-pretrained models on remote sensing related tasks. Prithvi was pretrained on HLS data for the contiguous U.S. using a transformer architecture. It was shown to perform well on flood mapping and burn scar segmentation. In our work, ImageNet, Satlas, and Prithvi are explored as pretraining methods as a way to apply transfer learning, an as a starting point before further training on the annotated DSWx dataset introduced in Section II-B.

B. Baselines for Masking

We describe the methods previously used to find water and cloud pixels from satellite data to compare the performance of our proposed model. Fmask and MNDWI are based on finding empirically derived thresholds to isolate cloud and water pixels. Deep learning methods [9], [29], [67] train a model to predict pixels similar to our proposed model. We discuss each of these as follows.

1) Fmask: The Fmask algorithm [10] uses the reflective bands and brightness temperatures from HLS to compute the probability of each pixel in the image being water or cloud. Optimal thresholds are empirically determined from a combination of the different bands and their ratios. The thresholds are fixed for reflective bands based on empirical results, whereas thresholds for thermal bands from HLS use the histogram of the image pixel values of the brightness temperatures.

2) Modified Normalized Difference Water Index: MNDWI [20] uses the green and SWIR-1 bands from HLS to capture water pixels. Equation (7) shows the pixel-wise operation to compute the probability of water pixels in the image. SWIR was used due to the observed higher absorption of water in this band [20]. At the same time, land tends to reflect SWIR light more than green light, resulting in lower MNDWI for nonwater areas

$$\text{MNDWI} = \frac{\text{green} - \text{SWIR}}{\text{green} + \text{SWIR}}. \tag{7}$$

From the computed values, the final water mask is determined by choosing an optimal static threshold t such that the mask is 1 for pixel values greater than t, and 0 otherwise. The threshold is chosen using the validation set. The same threshold is used for evaluation on the test set for comparison with other methods. The Otsu method [68] is also explored as a procedure to choose the threshold t.

3) Deep Learning Methods: LANA [9] is a recently released model for cloud and cloud shadow masking. Similar to Fmask, it uses the reflective bands and brightness temperatures to predict cloud masks. It learns cloud masks by utilizing a U-Net architecture (a type of CNN) with attention mechanisms incorporated in the skip connections between the encoder and decoder. Their proposed changes resulted in better performance compared to baseline methods Fmask [10] and U-Net Wieland [67]. In addition to the model, their work also introduced a collection of manually annotated satellite images from Landsat 8 for cloud and cloud shadow prediction. The labels come from USGS personnel annotations, the Spatial Procedures for Automated Removal of Cloud and Shadow project, and manually annotated tiles from Landsat. Their work resulted in 100 sets of annotated data from different global locations. LANA was trained on 99 out of the 100 sets and evaluated on the remaining set. This procedure was done five times, and the average was obtained and reported as the performance.

U-Net Wieland [67] was recently introduced in the literature for detecting cloud and cloud shadows, and was used as a baseline in LANA [9]. Similar to our model, it uses the visible, NIR, and SWIR bands. However, the architecture follows U-Net with an encoder, decoder, and skip connections. The model also outputs per-pixel labels to identify clouds and shadows. The model was trained using the specified train and test splits from LANA [9], and compared against our model for predicting clouds and cloud shadows.

DWM [29], [50] also uses six bands (visible, NIR, and SWIR bands) as input to predict a water mask. The model adopts a U-Net architecture. However, instead of having large feature maps in the early layers similar to the original U-Net, it was modified to use a constant feature map size throughout the network. Additional changes in the architecture were also applied to save memory such as increasing the stride in the convolution operation instead of using max pooling. For comparison, we train DWM on the DSWx train set for water mask prediction. We use the most updated released version of the model for comparison, while following the training procedure and parameters as closely as possible.

C. Performance Assessment

The models were evaluated based on the following:

1) masking performance;
2) efficiency;
3) accuracy of the downstream application to SSC estimation.

We discuss each of the evaluation methods as follows.

1) Masking: The performance for masking is computed through pixel-based metrics $F1$, recall, precision, and intersection over union (IoU) on the test set. The test set is composed of samples not seen by the model during training and validation. Recall (sometimes referred to as user's accuracy) is computed as the ratio of correctly predicted pixels to the total number of actual positive pixels in the label as shown in (8). Precision (also called producer's accuracy) is computed as the ratio of correctly predicted pixels to the total number of pixels that the model predicted as positive as shown in (9). $F1$ Score in (10) is the harmonic mean of recall and precision. Finally, IoU in (11) is the ratio of correctly predicted positive pixels to the union of the positive-labeled pixels and the positive-predicted pixels. In the equations below, TP is the true positive pixels, FN is the false negative pixels, and FP is the false positive pixels. For $F1$

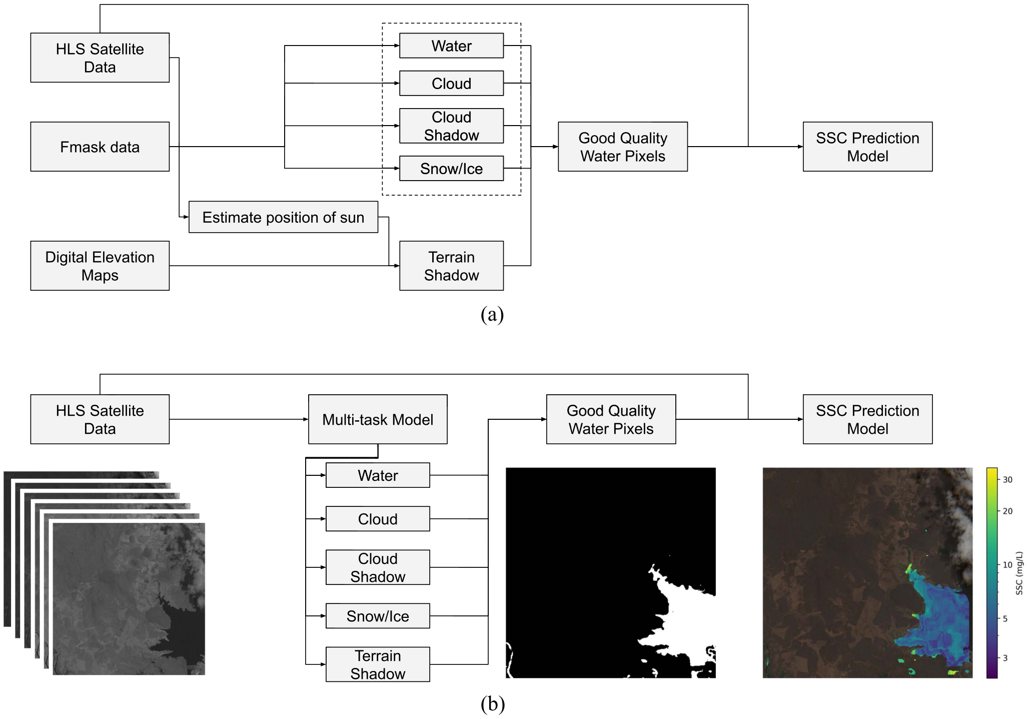

Fig. 3. *Pipeline for estimating suspended SSC.* (a) Standard SSC pipelines involve multiple inputs and several processing steps, which contribute to the memory and runtime requirements. (b) Our proposed pipeline only uses readily available HLS satellite images and estimates all masks faster by using a single multitask model. Using good quality water pixels by masking cloud, cloud shadow, snow/ice, and terrain shadow results in significantly better SSC estimates.

score, recall, precision, and IoU, better performing models would have higher corresponding quantities, with a maximum value of 100%

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{8}$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{9}$$

$$F1 \text{ Score} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{10}$$

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}. \tag{11}$$

2) Runtime, Memory, and Storage: In addition to masking performance, we also evaluate efficiency of the proposed multitask pipeline. Deploying on a global scale with frequent predictions requires a scalable model in terms of both runtime and memory consumption. We evaluate on both these characteristics, focusing on peak memory consumption and storage costs for the latter. The runtime is measured by aggregating the processing time of the different modules in Fig. 3. For comparisons between different methods, the runtime does not include downloading HLS data and assumes the HLS is already in memory.

Peak memory consumption would measure the required random access memory (RAM) requirement for running the pipeline. Storage costs are associated with the amount of space (e.g., in hard drive) to store auxiliary data to run the pipeline such as DEMs or pretrained models. Ideally, the proposed pipeline would require fewer resources (i.e., less total time and less memory) due to the introduction of the multitask model. The evaluations are done on AMD EPYC 7763 machine where 4 cores and 4 GB RAM are used for the process across 100 random samples.

3) Effect on Downstream SSC Estimation: The standard pipeline for SSC estimation from satellite images [see Fig. 3(a)] takes the HLS tiles, Fmask data, and DEMs as input. Water, cloud, cloud shadow, terrain shadow, and snow/ice masks are estimated to isolate good quality water pixels where the SSC is predicted. The terrain shadow mask is estimated by taking the metadata from HLS tiles to know the date, time, and location of the tile [69], [70], [71], [72]. From the metadata, the position of the sun is estimated. Taking the position of the sun and the topography of the area from the DEM, the approximate location of the terrain shadows can be calculated. The other masks required are then obtained as the outputs of Fmask algorithm, which are also accessible from the HLS project. The combination of these masks produce the good quality water pixels.

The good quality water pixels are then used as inputs to the SSC prediction model introduced in [54], which uses a two-stage machine learning model. The method uses an ensemble of two neural networks with the first network used for predicting low SSC values (0 to 20 mg/L), and the second network used for predicting middle to high SSC values (14 mg/L and above). Each network is composed of three fully connected layers, following the design of multilayer perceptrons [73]. Given a location (latitude, longitude) where the SSC is to be estimated, both models take as input the various statistics of the good quality water pixels within 300 m of the location. This includes the mean, median, standard deviation, minimum, and maximum for each HLS band. In addition, tiles identified with more than 30% cloud cover were not used for training.

The outputs of the two models are then combined based on two empirically determined SSC threshold values using the validation set. That is, if the output of the first model (predicting low SSC values) is below the first threshold, the output of the first model is used. Otherwise, we compare the second model's output (predicting middle to high SSC values) to the second threshold. If the output is above the second threshold, we take the second model's output. Otherwise, we take the average of the first and second model outputs. The two threshold values that give the lowest error on the validation set are used on the test set for model evaluation. The same threshold values are also used when deploying the model.

While the standard pipeline is straightforward and uses readily available data, it involves several processing steps and could benefit from not just improving the accuracy of isolating good quality water pixels, but also from streamlining the process. Using a single multitask model, we can take only HLS data as input and predict multiple masks at the same time. Memory is saved by limiting the input data required to just HLS reflective bands. Run time is also significantly reduced since all the processing is done by a single model. Fig. 3(b) shows the proposed pipeline. While we show the pipeline specific to SSC estimation, similar frameworks exist for other applications, and can similarly be optimized by introducing a multitask masking module.

To further evaluate our framework, we compare the performance of downstream application of SSC predictions when using the standard pipeline without a multitask model, and the optimized pipeline with the multitask model. Ideally, we expect the performance of the SSC model to either stay the same or improve with the use of the multitask model. We use the metrics RMSE [see (12)], MAE [see (13)], and bias [see (14)] on the predicted SSC values to evaluate the performance of the SSC model across the different pipelines. We also report various statistics of the absolute error $|y_i - \hat{y}_i|$ for all i in the test dataset—we include the median, maximum, minimum, and standard deviation. In the following equations, the metrics are computed across N samples, where y_i is the ground truth or SSC label, and $\hat{y}_i$ is the predicted SSC for sample i. An optimal model would have values close to zero for the following metrics:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum\nolimits_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{12}$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{13}$$

$$\mathrm{bias} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)\,. \tag{14}$$

IV. Results

A. Masking Performance

1) Comparison Between Multitask and Single-Task Models: We compare the quantitative performance of the multitasking model to the single-task counterparts. The single-task models use five separate models, where one model is trained on only one task [see Fig. 2(a)]. The multitask model is trained on all five tasks simultaneously [see Fig. 2(b)]. Table III shows that the performance of the multitask models is either similar or better than the single-task models. Three different architectures that use CNN (DeepLabv3+, MobileNetv3) and transformer (Swin-T) were evaluated. Most of the improvement from using multitask models can be observed on snow/ice masking when using CNN-based models DeepLabv3+ and MobileNetv3 with more than 10% metric improvement.

2) Comparison of Multitask Models Across All Masks: All five masks predicted by the multitask models were evaluated on the held out DSWx test set. Fig. 4 shows qualitative results from training a DeepLabv3+ multitasking model across the five masks. The results show DeepLabv3+ can successfully identify water, cloud, cloud shadow, snow/ice, and terrain shadow simultaneously. Table IV shows the $F1$ score across the different masks for various architectures, pretraining methods (transfer learning), and model types. The best performance is in bold, while the second best is underlined. While all multitask models can reasonably predict the different masks, Swin-T pretrained on Satlas demonstrates superior performance compared to other architectures. While the architecture itself plays a role in higher accuracy due to its larger capacity and global feature representations, the pretraining method also plays a significant role in improving the performance. When comparing the performance of the same architecture (Swin-T) pretrained on ImageNet and pretrained on Satlas, the latter version has as much as 30% $F1$ score improvement for cloud shadow masking. We additionally find that not applying transfer learning at all reduces performance. For a DeepLabv3+ model that is not pretrained, water masking on the test set resulted in 86.04% $F1$ score, almost a 4% decrease in performance from the 89.67% $F1$ score of the same model pretrained with ImageNet.

3) Water Mask Comparison of Multitask Models With Baselines: In addition to the performance of simultaneously predicting all masks, the performances of individual masks were also compared and evaluated. Table V shows the performance of the different methods for water masking using the DSWx test set. The best performance is in bold, while the second best is underlined. In the evaluation, only the water mask output of the multitask model was used; other masks were discarded. The baseline method DWM [50] was trained on the train set of DSWx and only predicts water masks. It was trained with the same parameters and setup recommended in their paper. The baseline MNDWI was evaluated out of the box for fair comparison, resulting in a 16.62% $F1$ score. However, standard practice in hydrology has moved beyond straightforward application of MNDWI. To achieve results similar in practice, we apply cloud and shadow masking from DSWx and remove predictions over regions with no data. We include these results in

TABLE III
MASKING PERFORMANCE OF THE MULTITASK MODEL COMPARED TO THE SINGLE-TASK MODELS ON DSWX TEST SET (HIGHLIGHTED CELLS INDICATE BETTER METRICS WHEN USING THE MULTITASK MODEL COMPARED TO ITS SINGLE-TASK EQUIVALENT)

DeepLabv3+ (ImageNet pre trained)

Label	**Single task models**				**Multi task model**			
	$F1$ Score	Precision	Recall	IoU	$F1$ Score	Precision	Recall	IoU
Water	89.55%	86.99%	92.26%	81.07%	89.67%	87.91%	91.50%	81.27%
Cloud shadow	61.94%	64.39%	59.66%	44.86%	60.16%	59.63%	60.69%	43.02%
Cloud	90.12%	89.00%	91.27%	82.02%	87.89%	90.21%	85.68%	78.39%
Snow/ice	60.31%	51.81%	72.15%	43.18%	70.53%	63.99%	78.56%	54.48%
Terrain shadow	97.33%	95.26%	99.49%	94.79%	97.05%	94.62%	99.61%	94.27%

MobileNetv3 (ImageNet pre trained)

Label	**Single task models**				**Multi task model**			
	$F1$ Score	Precision	Recall	IoU	$F1$ Score	Precision	Recall	IoU
Water	88.03%	84.47%	91.91%	78.63%	88.18%	85.16%	91.42%	78.86%
Cloud shadow	59.44%	59.44%	59.43%	42.29%	58.70%	58.58%	58.82%	41.54%
Cloud	90.27%	90.10%	90.44%	82.26%	89.96%	90.25%	89.67%	81.75%
Snow/ice	59.50%	52.78%	68.19%	42.35%	74.31%	72.33%	76.41%	59.13%
Terrain shadow	97.30%	95.16%	99.53%	94.74%	97.13%	94.83%	99.55%	94.43%

Swin-T (Satlas pre trained)

Label	**Single task models**				**Multi task model**			
	$F1$ Score	Precision	Recall	IoU	$F1$ Score	Precision	Recall	IoU
Water	89.77%	87.35%	92.31%	81.43%	91.10%	90.62%	91.58%	83.65%
Cloud shadow	64.64%	63.78%	65.52%	47.75%	63.10%	63.44%	62.75%	46.09%
Cloud	92.59%	91.88%	93.31%	86.20%	92.42%	93.14%	91.72%	85.91%
Snow/ice	75.83%	72.70%	79.23%	61.06%	78.09%	73.92%	82.76%	64.06%
Terrain shadow	97.50%	95.70%	99.37%	95.12%	97.26%	95.19%	99.44%	94.68%

TABLE IV
$F1$ SCORE FOR VARIOUS MULTITASKING MODELS ACROSS MASK TYPES ON THE DSWX TEST SET

Model	Pre-training	Model type	Water	Cloud shadow	Cloud	Snow/Ice	Terrain shadow
DeepLabv3+	ImageNet	CNN	89.67%	60.16%	87.89%	70.53%	97.05%
MobileNetv3	ImageNet	CNN	88.18%	58.70%	89.96%	74.31%	97.13%
SegNet	ImageNet	CNN	83.47%	58.57%	89.36%	71.11%	97.01%
ResNet50	Satlas	CNN	81.33%	53.27%	87.92%	64.84%	96.84%
Swin-T	Satlas	Transformer	**91.10%**	**63.10%**	**92.42%**	**78.09%**	**97.26%**
Swin-T	ImageNet	Transformer	80.73%	35.81%	85.30%	71.36%	96.74%
ViT-B/16	ImageNet	Transformer	82.56%	37.38%	85.89%	70.85%	96.74%
ViT-B/16	Prithvi	Transformer	76.61%	30.87%	82.36%	69.76%	96.74%

TABLE V
PERFORMANCE OF THE VARIOUS METHODS AND BASELINES FOR WATER MASKING

Method	Pre training	Model type	$F1$ Score (↑)	Precision (↑)	Recall (↑)	IoU (↑)
MNDWI			58.43%	78.92%	46.39%	41.28%
DWM		CNN	82.21%	78.54%	86.24%	69.79%
DeepLabv3+	ImageNet	CNN	89.67%	87.91%	91.50%	81.27%
MobileNetv3	ImageNet	CNN	88.18%	85.16%	91.42%	78.86%
SegNet	ImageNet	CNN	83.47%	82.94%	84.01%	71.63%
ResNet50	Satlas	CNN	81.33%	78.76%	84.08%	68.54%
Swin-T	Satlas	Transformer	**91.10%**	**90.62%**	**91.58%**	**83.65%**
Swin-T	ImageNet	Transformer	80.73%	77.88%	83.80%	67.69%
ViT-B/16	ImageNet	Transformer	82.56%	81.15%	84.03%	70.30%
ViT-B/16	Prithvi	Transformer	76.61%	74.60%	78.74%	62.09%

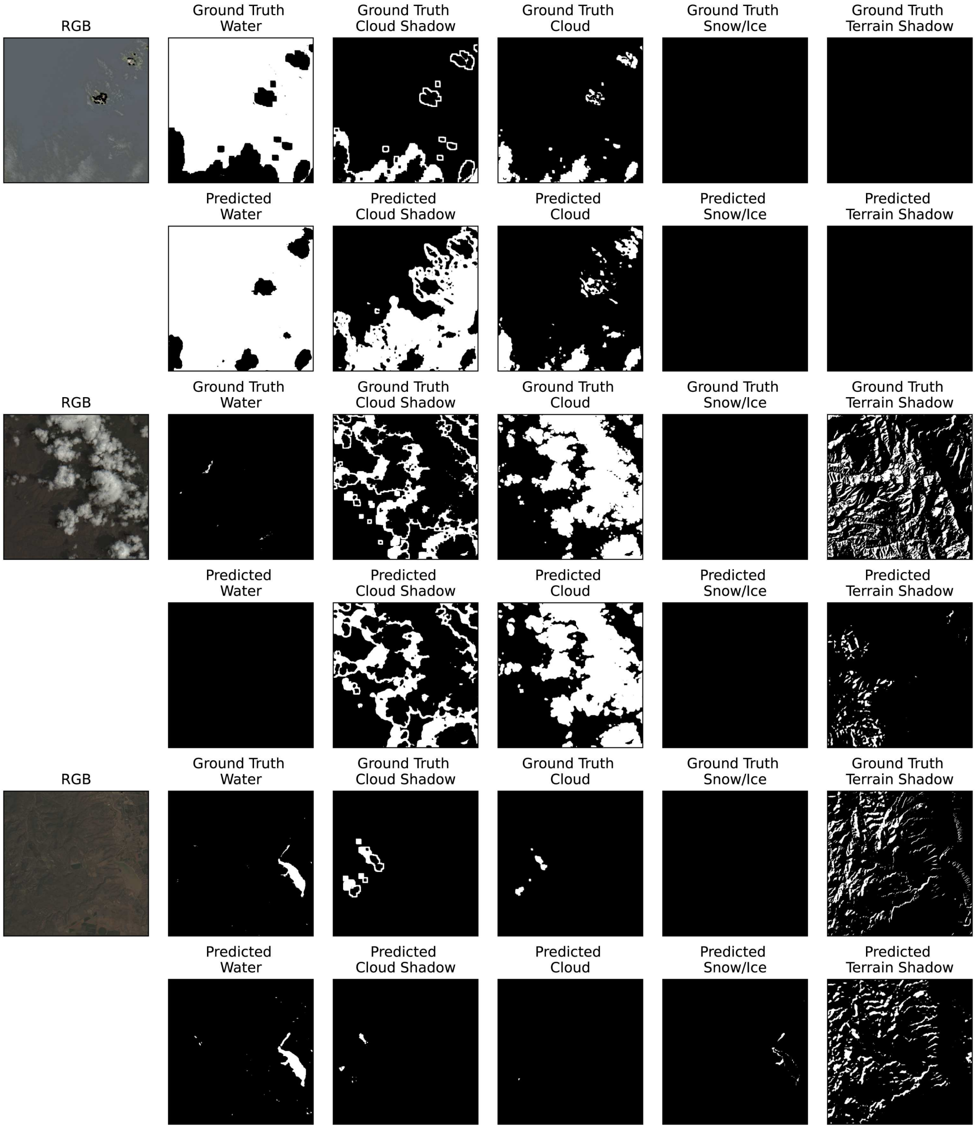

Fig. 4. *DeepLabv3+ multitasking model results on three samples from OPERA DSWx test set.* The RGB images are shown together with the corresponding ground truth and predicted masks. White pixels denote the presence of the mask, and black pixels otherwise. The model predictions across different types of masks closely match the ground truth based on DSWx.

Table V, resulting in a 58.43% $F1$ score. Other methods were not applied with the same cloud and shadow masking, yet the results still show multitask models Swin-T (Satlas pretrained) and DeepLabv3+ (ImageNet pretrained) have the best performance when masking water pixels with 91.10% and 89.67% $F1$ score, respectively. MobileNetv3 also has competitive performance against other models. The multitask models introduce around 10% $F1$ score improvement compared to baseline methods DWM and MNDWI.

Fig. 5 shows results of predicting the water mask across different methods. The top 3 best performing multitask models are displayed and compared against the baselines. The visual results show that MNDWI is sensitive high reflectance values, which leads to pixels that are falsely identified as water. It also frequently fails to identify water pixels. DWM tends to predict smooth water boundaries, and can fail to capture fine details such as branching out in rivers. Due to the multitasking setup of the proposed model, the outputs are less sensitive to clouds and shadows, since the model is trained to identify all the different types of masks at the same time. At the same time, the multitask models are able to capture fine details in the water boundaries.

4) Cloud Mask Comparison of Multitask Models With Baselines: The cloud masking performance of the multitask models were also evaluated on both the DSWx test set, and the LANA [9] benchmark. The evaluation on the LANA benchmark involved training the multitask models on the sets of data not used for evaluation, and evaluated on five unseen sets of data, similar to the evaluation outlined in their paper for fair comparison. The dataset contains manually labeled cloud pixels from USGS personnel. Table VI shows the difference in performance between the different multitask models and the baselines LANA, Fmask, and U-Net Wieland. The best performance is in bold, while the second best is underlined. The baselines are models

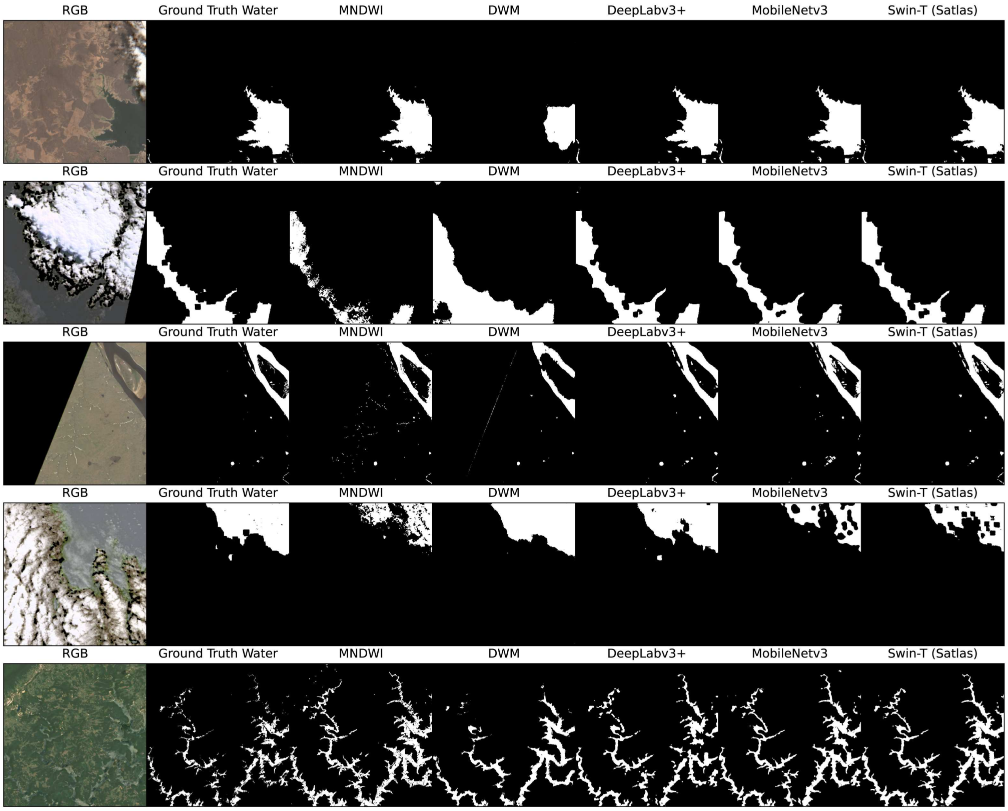

Fig. 5. *Water masking results of different methods.* MNDWI results shown here were filtered to remove clouds and shadows, while results from other methods were not filtered in any way. MNDWI fails to identify several water pixels, while DWM tends to predict smooth water boundaries which could miss details. DeepLabv3+, MobileNetv3, and Swin-T outperform the baselines but DeepLabv3+ is more robust to noise (see row four in the figure).

TABLE VI
PERFORMANCE OF CLOUD MASKING ON MANUALLY LABELED LANA [9] DATASET

	Model	Cloud (↑)	Cloud Shadow (↑)	Clear (↑)
Baselines	LANA [9]	92.42%	57.53%	89.02%
	Fmask [10]	89.81%	45.42%	88.09%
	U-Net Wieland [67]	87.68%	52.06%	86.19%
Multi-task Models	DeepLabv3+ (ImageNet pretrained)	92.64%	65.79%	95.54%
	MobileNetv3 (ImageNet pretrained)	**93.70%**	63.60%	95.77%
	SegNet (ImageNet pretrained)	91.19%	57.64%	95.19%
	ResNet50 (Satlas pretrained)	85.78%	63.67%	92.77%
	Swin-T (Satlas pretrained)	92.96%	**69.56%**	**95.80%**
	Swin-T (ImageNet pretrained)	82.73%	4.32%	92.49%
	Vit-B/16 (ImageNet pretrained)	59.89%	0.01%	88.16%
	Vit-B/16 (Prithvi pretrained)	81.38%	6.94%	91.52%

developed specifically for cloud and cloud shadow masking. Although the proposed multitask models are using only 6 bands from the Landsat data—as opposed to LANA and Fmask that use 8 bands for prediction—the performance is similar (for cloud masking), or even better (for cloud shadow and clear masks) than the baselines.

As additional reference, a previous method of cloud and cloud shadow detection on Landsat data [30] explored different CNN models and compared against a variant of Fmask (CFmask). They obtained around 94% overall accuracy for detecting cloud-based classes (cloud, cloud shadow, thin cloud, clear), and improved over CFmask accuracy by around 10%. However, their sampling of Landsat data into training, validation, and test sets differs from our setup since they sample their data at the patch-level, where each training patch is a small crop of the larger Landsat image scene (and could potentially have an overlap with another patch in the validation/test set), similar to [74], [75], [76]. In contrast, our training uses data sampled at the scene-level, ensuring that training and testing patches do not come from the same image and have no spatial overlap, similar

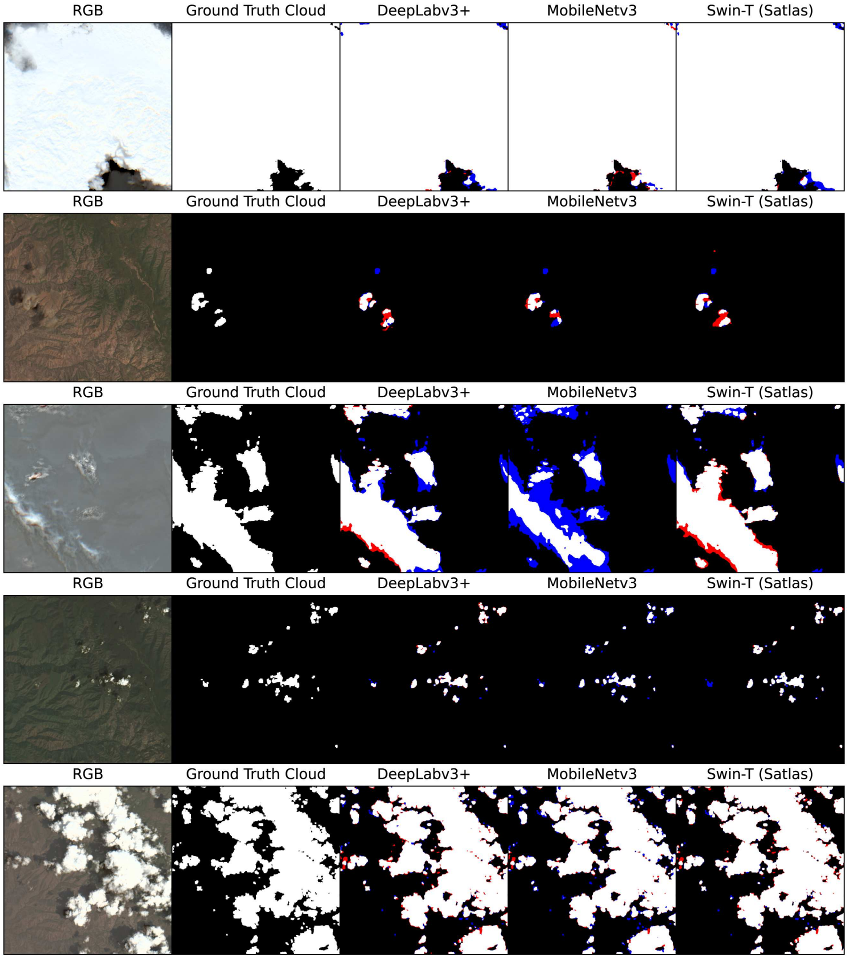

Fig. 6. *Cloud masking results for different methods on the test set.* False positive pixels are colored red, and false negatives are colored blue. While all models perform similarly, MobileNetv3 tends to miss more true positives, i.e., it incorrectly classifies could pixels as noncloud.

to LANA [9]. We do this to avoid inflating the reported metrics since patches from the same scene could have similar cloud and surface conditions.

Figs. 6 and 7 show additional results for predicting cloud and cloud shadow pixels of the multitask models on the DSWx test set. These model results were based on training on the DSWx train set. The results show that all three models DeepLabv3+, MobileNetv3, and Swin-T can accurately predict both clouds and cloud shadows when compared to the ground truth labels, as supported by the results in Table IV. There are slight differences in MobileNetv3 and DeepLabv3+, where MobileNetv3 would sometimes miss true positive pixels, and DeepLabv3+ would sometimes have false positive pixels. However, overall, the performance of the models are comparable.

5) Additional Qualitative Results on Other Masks: Figs. 8 and 9 show qualitative results on predicting terrain shadow and snow/ice, respectively. The three best performing multitask models are visualized with the RGB bands and the label for each of the samples. Looking at the visual results in Fig. 8 for terrain shadow, Swin-T pretrained on Satlas have closely aligned outputs with the ground truth, where it is able to capture finer details and even predict parts partially occluded by clouds. DeepLabv3+ terrain predictions are more sensitive to cloud occlusions in comparison. This supports the reported $F1$ score in Table IV for terrain shadow where Swin-T slightly outperforms DeepLabv3+ and MobileNetv3.

Snow/ice pixel identification results in Fig. 9 show DeepLabv3+ being able to identify more snow/ice pixels, but it tends to exceed the boundaries and falsely predict surrounding pixels as snow/ice, and thus, has more false positives. MobileNetv3 and Swin-T have less false positives and while the predictions show less pixels identified as snow/ice, the positive predictions appear in accurate locations. As a result, MobileNetv3 and Swin-T have higher quantitative metrics on snow/ice masking (see Table IV).

B. Effect on Downstream Application: SSC Estimation

1) Time: Table VII shows comparisons on the amount of time to process a single sample for an SSC model, and the amount of time to process 400 k samples.The best performance with the lowest amount of processing time is in bold, while the second best is underlined. On average, there are 400 k samples per day released as part of HLS based on the data from EOS-DIS. Assuming a linearly increasing runtime based on a single sample, processing 400 k samples would take approximately

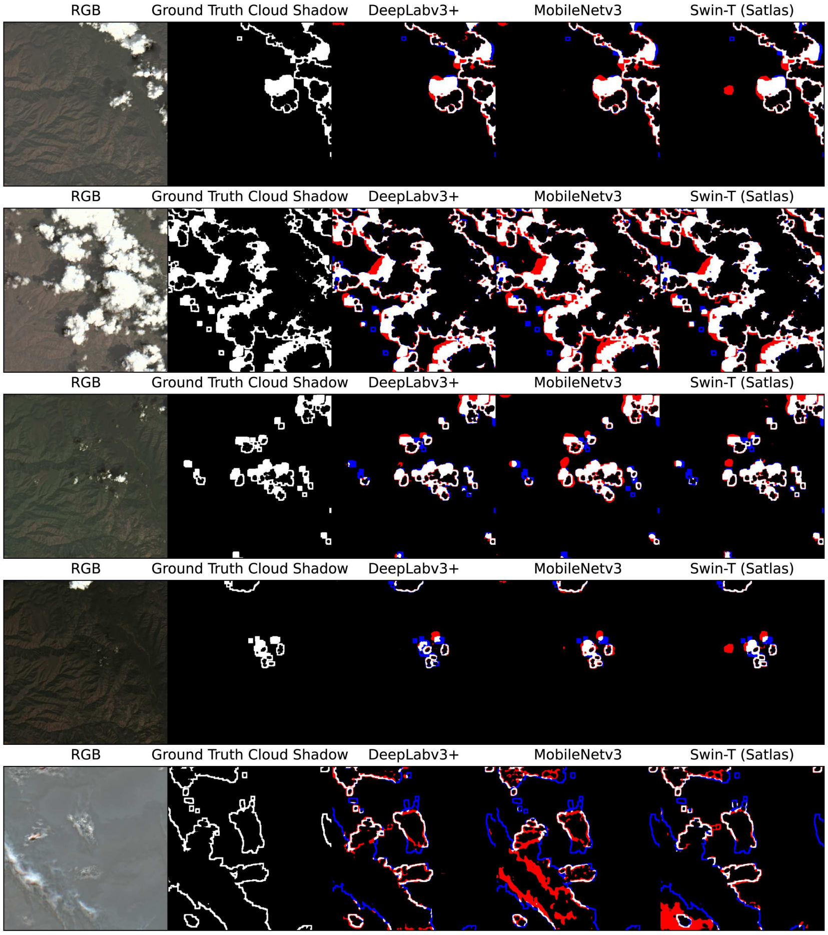

Fig. 7. *Cloud shadow masking results for different methods on the test set.* False positive pixels are colored red, and false negatives are colored blue. The performance of the models on cloud shadows is similar to cloud masking, and show that the models have comparable results.

TABLE VII
Runtime Comparison of Standard SSC Pipeline and the Proposed Multitask SSC Pipeline on 4 Cores of an AMD EPYC 7763 Machine

	Runtime on 1 sample (s)	Runtime on 400k samples (days)	Improvement (%)
Standard SSC Pipeline	18.757	86.84	-
DeepLabv3+ (ImageNet pretrained)	2.002	9.27	89.33%
MobileNetv3 (ImageNet pretrained)	**0.601**	**2.78**	**96.80%**
SegNet (ImageNet pretrained)	2.259	10.46	87.96%
ResNet50 (Satlas pretrained)	7.209	33.38	61.57%
SwinT (Satlas pretrained)	6.260	28.98	66.62%
SwinT (ImageNet pretrained)	1.254	5.80	93.32%
ViT-B/16 (ImageNet pretrained)	2.450	11.34	86.94%
ViT-B/16 (Prithvi pretrained)	3.493	16.17	81.38%

86.84 days when using the standard SSC pipeline. In contrast, using a MobileNetv3 multitask model would result in only 2.78 days for 400 k samples, a 30× speedup (or 96.80% runtime improvement). Table VIII shows how each module contributes to the amount of time to process a single sample. Majority of the runtime in a standard SSC pipeline comes from combining masks from different sources to get good quality water pixels. Unlike the standard SSC pipeline that requires reprojections and alignments due to the different sources of data, our proposed pipeline predicts masks that can be directly combined without additional processing overhead. Four cores of an AMD EPYC 7763 was used to evaluate the runtime and memory.

2) Memory and Storage: Table IX shows the difference in memory consumption between the standard and the proposed multitask pipelines. The lowest memory consumption is in bold. The peak memory consumption is measured by running the

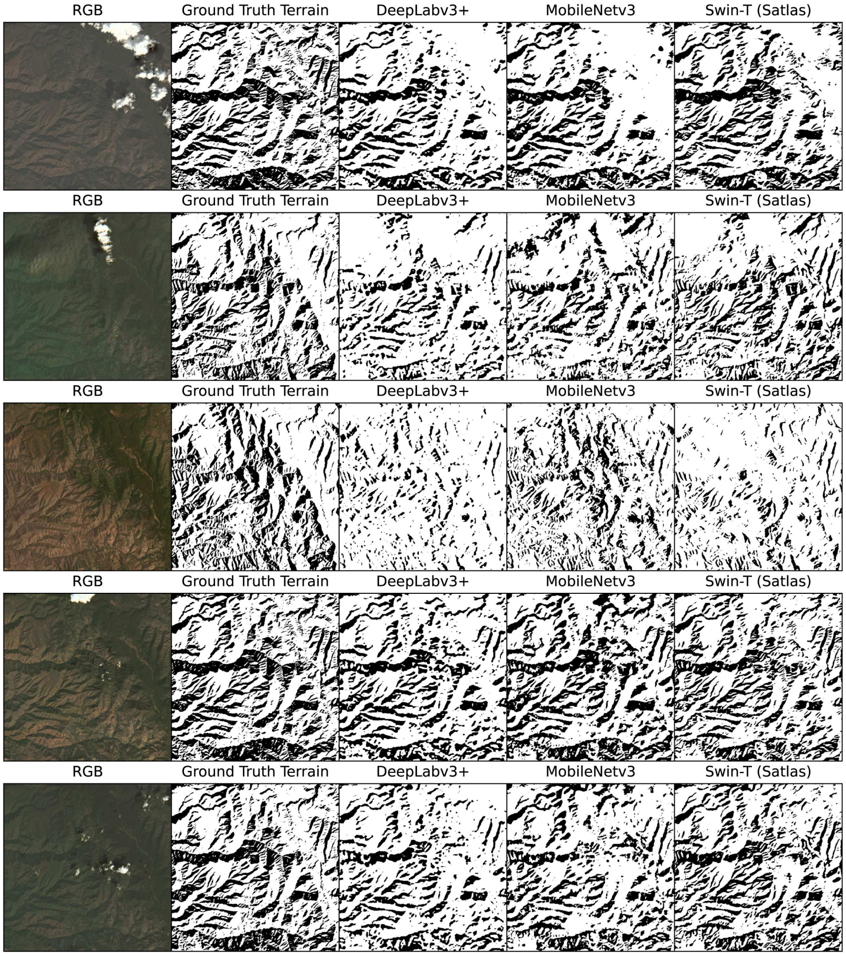

Fig. 8. *Terrain shadow masking results of different methods on the test set.* DeepLabv3+ is more sensitive to clouds and its surrounding pixels, while MobileNetv3 and Swin-T can still predict near clouds.

TABLE VIII
RUNTIME BREAKDOWN OF A STANDARD SSC PIPELINE AND OUR PROPOSED MULTITASK SSC PIPELINE WITH DEEPLABV3+ ON 4 CORES OF AN AMD EPYC 7763 MACHINE FOR A SINGLE SAMPLE

	Standard SSC pipeline	Time (s)	Proposed SSC pipeline	Time (s)
Pre-processing	Estimate position of sun	2.05	-	-
Obtaining masks	Water Cloud Cloud shadow Snow/ice	1.94	Water Cloud Cloud shadow Snow/ice	1.96
	Terrain shadow	5.61	Terrain shadow	
Combining masks	Good quality water pixels	9.10	Good quality water pixels	0.04

pipelines to produce features for the SSC model. The standard SSC pipeline requires at least 1.85 GB of RAM. However, when using MobileNetv3 as the multitask model in the proposed pipeline, the required RAM reduces to 0.833 GB, less than half the requirement of the standard pipeline. The other models, DeepLabv3+ and Swin-T, also require less RAM than the standard pipeline. At the same time, the storage costs associated with any of the multitask models are about a hundredth of the standard SSC pipeline, since the proposed pipeline would only need to store the pretrained models. The standard pipeline requires storing all DEMs covering all landmasses, which is 96 GB in size, while the proposed multitask pipeline only requires the models themselves to be stored, which is around 1 GB in size.

3) Accuracy: The effect of using a more accurate multitask model was also evaluated on the downstream application of SSC estimation. DeepLabv3+ pretrained on ImageNet was used as the multitask model. While MobileNetv3 is faster, DeepLabv3+ showed a better performance for water masking and is

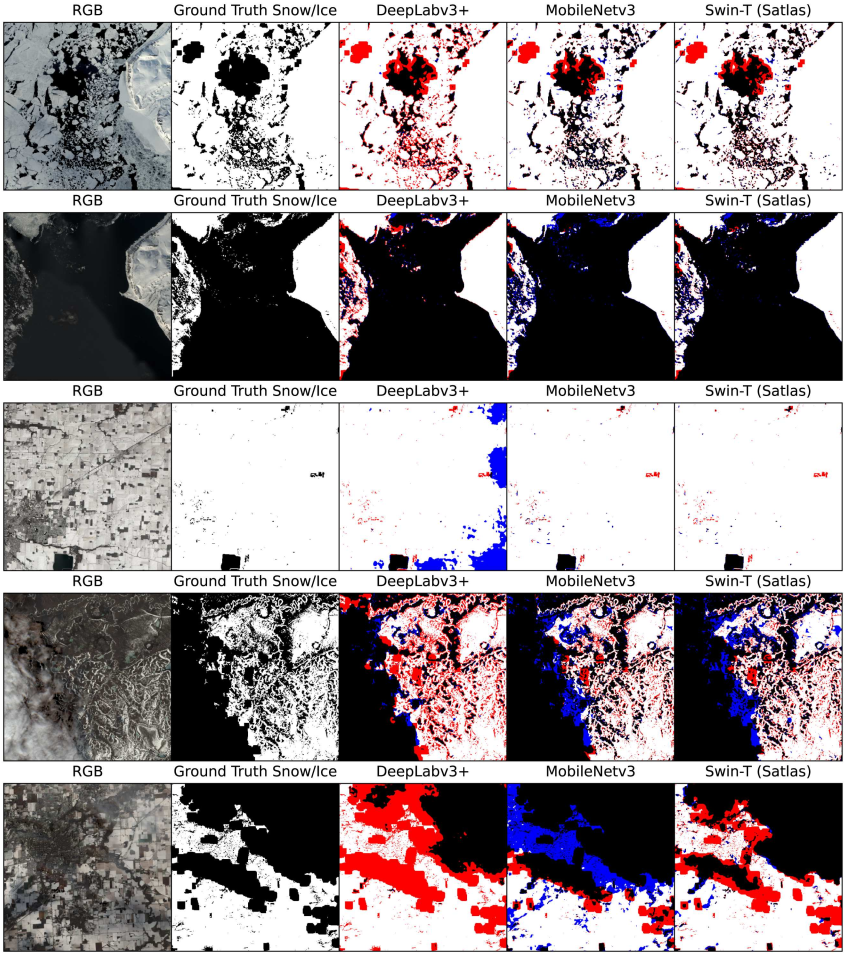

Fig. 9. *Snow/ice masking results of different methods on the test set.* Results from MobileNetv3 and Swin-T are able to capture true positive pixels for snow/ice. While DeepLabv3+ classifies more pixels as snow/ice, there are more false positives (red pixels) present. On the other hand, MobileNetv3 has more false negatives (blue pixels).

TABLE IX
PEAK MEMORY CONSUMPTION AND STORAGE OVERHEAD COMPARISON BETWEEN THE STANDARD AND PROPOSED MULTITASK SSC PIPELINES

	Peak memory Consumption (GB)	Storage Overhead (GB)
Standard SSC Pipeline	1.851	96.000
DeepLabv3+ (ImageNet pretrained)	1.589	0.257
MobileNetv3 (ImageNet pretrained)	**0.833**	**0.014**
SegNet (ImageNet pretrained)	1.311	0.288
ResNet50 (Satlas pretrained)	1.179	0.304
Swin-T (Satlas pretrained)	1.290	0.343
Swin-T (ImageNet pretrained)	1.235	0.361
ViT-B/16 (ImageNet pretrained)	2.072	1.000
ViT-B/16 (Prithvi pretrained)	2.625	1.260

smaller than Swin-T pretrained on Satlas. Table X shows the performance of the SSC model when trained on features produced by the standard SSC pipeline using Fmask and DEM compared to the SSC model when trained on features produced by the optimized multitask pipeline. We report the statistics of the error on the test set of the dataset. The model from [54] for SSC estimation was trained and evaluated as outlined in their methodology. The results show that the performance of the SSC model either improved or stayed similar across the metrics when using the proposed multitask pipeline. The RMSE between ground truth and predicted SSC values reduced by 2.64 mg/L. There is also significant reduction in the 95th (E95) and 90th (E90) percentile error with 13.55 mg/L and 6.25 mg/L error reduction, respectively. This can be attributed to better quality

TABLE X
SSC MODEL PREDICTION METRICS USING THE STANDARD SSC PIPELINE AND THE PROPOSED MULTITASK PIPELINE USING A TWO-STAGE ML MODEL (↓ MEANS LOWER IS BETTER)

SSC prediction metrics (mg/L)			
Metric	Standard SSC Pipeline	Multitask SSC Pipeline	Δ(↓)
RMSE (↓)	60.61	57.97	– 2.64
MAE (↓)	24.44	22.88	– 1.56
Bias (↓)	22.53	21.23	– 1.30
Max Abs Error (↓)	701.05	700.40	– 0.65
E95 Abs Error (↓)	119.22	105.67	– 13.55
E90 Abs Error (↓)	63.86	57.61	– 6.25
E75 Abs Error (↓)	18.26	16.60	– 1.66
Median Abs Error (↓)	5.73	5.34	– 0.39
Std Dev Abs Error (↓)	55.96	52.97	– 2.99
Min Abs Error (↓)	0.00	0.02	+ 0.02

water pixels isolated for training the SSC model, as produced by the multitask model. Fmask, the water mask used in the standard SSC pipeline, tends to falsely identify areas of water, which could contribute to incorrect features being used to estimate SSC values.

V. Discussion

Classifying pixels from satellite images for downstream applications on a global level requires both accurate and efficient frameworks. Evaluation should also be done on the full end-to-end pipeline to capture the positive effects, if any, of the introduced masking. SSC estimation for all rivers in all landmasses, for example, relies on good quality water pixels not occluded by clouds, cloud shadows, terrain shadows, and snow/ice. At the same time, frequent analysis of SSC in global surface waters would only be possible with pipelines that produce these masks at a fast rate. A standard SSC pipeline uses the Fmask algorithm on HLS and DEMs to isolate good quality water pixels, but the reprojections and alignments needed to process information from different data sources introduce compute overhead. In addition, Fmask has limitations for accurately isolating water pixels due to its dependence on empirically derived thresholds. We introduced a more efficient framework that uses a multitask model to predict all necessary masks for SSC estimation while illustrating that its improved masking performance also results in better performance for SSC estimation.

A. Masking Performance

We showed through our experiments that the multitask model can benefit from learning different masks in a consistent manner. In the DSWx dataset, since all five masks are predicted together, the multitask model has context on where clouds are, and can learn to predict other masks such as water and terrain shadow that are not occluded by clouds. This advantage is apparent for the water and snow/ice masks that show significant improvement when trained with the multitask model over the single-task model equivalents (see Table III). There are especially large improvements for snow/ice in particular for CNN models DeepLabv3+ and MobileNetv3 with more than 10% improvement on all metrics. This could be due to the smaller number of positive samples for snow/ice globally, which limits the accuracy of the single-task model. In contrast, the ability of the multitask model to identify other types of masks (e.g., water) can significantly help the model identify, which ones are not snow/ice and reduce false positives as observed in the increase in precision for the multitask models. The advantage is further supported by the higher accuracy in both water mask (see Table V and Fig. 5) and cloud mask (see Table VI) experiments compared to existing models LANA, Fmask, U-Net Wieland, and DWM.

Performance of both single-task and multitask models are relatively lower for snow/ice and cloud shadow masks in DSWx compared to other masks such as water and cloud masks (see Table III). This is expected due to the smaller number of positive samples the models can learn from as reported in Table II. These classes are also harder to classify due to ambiguity with other classes, such as clouds that could be mistaken for cloud shadow and vice versa. Similar phenomena could also be observed for snow/ice and clouds. In contrast, there are a large number of positive samples for cloud and terrain shadow, where we see better performance across all models (around 90% $F1$ score). In addition to limited snow/ice samples that the model can learn from, it should also be noted that while DSWx provides sufficient labels for snow/ice, there are occasional misclassifications. These typically occur over waters with unusual colors due to high sediment concentrations or dissolved solids (Jones, 2019). It is possible that some disagreements between model predictions and labels come from erroneous labels. Examining manual cloud annotations from LANA, and comparing them against DSWx labels, we find that cloud labels from DSWx have an $F1$ score of 89.81%.

Experiments on the LANA dataset (see Table VI) show that predicting cloud shadow is more challenging than clouds or clear pixels due to the smaller number of samples available. Clouds and clear pixels are also easier to identify due to their distinct features. However, there is at least a 10% $F1$ score improvement in the prediction of cloud shadow when using the multitask model Swin-T (pretrained with Satlas) when compared with LANA, Fmask, or U-Net Wieland. Similar to previous results, DeepLabv3+ and MobileNetv3 also show competitive results with Swin-T. While Swin-T performs well, other transformer architectures pretrained with ImageNet or Prithvi have significantly lower performance on the LANA dataset as compared to the DSWx dataset. This is due to the small number of data samples used for training; the LANA dataset only contains around 16 000 training samples as opposed to the DSWx dataset with more than 80 000 training samples.

Although DSWx labels are also generated by an algorithm, DSWx requires data from Copernicus DEM, Copernicus land cover, ESA worldcover, NOAA GSHHS shapefile, and HLS [45]. The workflow for generating DSWx labels involve applying multiple steps to the aforementioned data sources such as filtering, and conducting diagnostic tests to produce masks. While their method is sufficiently robust for detecting water, clouds, and shadows, applying it on large amounts of data (e.g., decades of historical data from Landsat and Sentinel) would be a difficult task. Our work introduces a promising framework that runs fast using only HLS data, making it possible to run even on past satellite images. In addition, we show that our framework can be used on manually labeled data from LANA [9] (see Table VI). Even when evaluating on datasets with manually labeled data, we show our method's generated masks perform better than current state-of-the-art methods. These results are encouraging and offer an interesting way forward for operational production of OPERA data: the huge effort required to generate DSWx can be used and honored with a more efficient and likely more accurate (via LANA) representation of itself, potentially saving resources for the operational production of OPERA.

B. SSC Estimation End-to-End Pipeline Efficiency

The standard SSC pipeline had a measured processing of 86.84 days, which is impractical when running almost daily SSC predictions for in-depth global surface water analyzes (see Table VII). As an alternative, multiple machines can be used in parallel to reduce processing time. However, this would require significant resources and result in larger costs. The large memory overhead required for a standard pipeline (see Table IX) also contribute to the impracticality of increasing the hardware scale to reduce processing time. Table VIII shows the breakdown of the runtime and how different modules contribute to the processing time. The bulk of the processing time comes from aligning the different masks to isolate good quality pixels. Since different masks come from different sources (e.g., DEM or Fmask), the projection could be different, requiring warping and reprojections to be applied.

With the proposed multitask pipeline, all the reprojections that come from using different data sources are eliminated. The processing time can be reduced by as much as 96.80%, such that 400 k samples can be processed in less than 3 days on a 4-core CPU (see Table VII). This approach would require fewer machines in parallel, if necessary at all. MobileNetv3+ has the fastest runtime of 2.78 days for 400 k samples, with Swin-T pretrained with ImageNet and DeepLabv3+ having runtimes of 5.80 and 9.27 days, respectively.

The multitask pipeline also has comparably lower requirements for memory and storage compared to the standard pipeline (see Table IX). When running multiple parallel processes for obtaining features for SSC models, this could significantly reduce costs. In addition to efficiency improvements, it was shown that our proposed multitask pipeline improves SSC estimation accuracy (see Table X). Taking the improvements in performance, runtime, and memory consumption, the multitask pipeline presents a promising framework for downstream applications such as global SSC prediction.

C. Performance of Different Architectures and Pretraining

Experiments discussed in this work show that using multitask models for simultaneously predicting water, cloud, cloud shadow, terrain shadow, snow/ice result in a better performance over its single task equivalent while requiring less training resources (see Fig. 2). Different architectures MobileNetv3, DeepLabv3+, and Swin-T also show larger performance improvements when using a multitask model for predicting labels with small positive samples (e.g., snow/ice). Our experiments have shown that a multitask model using Swin-T pretrained with Satlas have superior masking accuracy compared to baselines and other multitask models using a different architecture. This was illustrated on different types of masks. For water masking, Swin-T performed better than Fmask [10], MNDWI [20], and DWM [50] with at least a 10% increase in $F1$ score. Swin-T also outperformed other baselines Fmask, LANA, and U-Net Wieland on cloud masking on the LANA-introduced dataset, where Swin-T has 10%, 12%, and 6% $F1$ score improvement for cloud, cloud shadow, and clear pixel identification, respectively (see Table VI). More generally, Swin-T pretrained on Satlas was also shown to perform better than other architectures for predicting five masks simultaneously (see Table IV).

Our experiments also show the effect of transfer learning through various pretraining datasets. CNN models DeepLabv3+ and MobileNetv3, both ImageNet pretrained, show competitive masking performance when compared to Swin-T, with only around 3% $F1$ score difference across most masks (see Table IV). Despite also being pretrained on ImageNet, transformer models Swin-T and ViT-B/16 have lower $F1$ score than their CNN counterparts. This could be attributed to the lack of inductive bias in transformers, requiring models to be trained with larger datasets (e.g., Satlas) to take advantage of the global representation learning. While ImageNet is considered a large dataset with 1 million training images, Satlas is around 100× larger. At the same time, Satlas is curated for remote sensing data, but ImageNet is a general dataset that covers multiple objects (e.g., animals, musical instruments, plants) that are not typically seen from satellite images.

While Swin-T (pretrained with Satlas) performs well on simultaneously predicting water, cloud, cloud shadow, terrain shadow, and snow/ice, it should be noted that it is also larger and slower than other architectures (see Tables VII and IX). Depending on the application, it would be advantageous to also consider a multitask MobileNetv3 model, which is almost as good as Swin-T in terms of masking accuracy, but only runs for a sixth of the time required for Swin-T, and consumes less RAM and storage. DeepLabv3+ can also be another choice, which can perform more accurately than MobileNetv3, but also consumes less RAM and storage than Swin-T. Our work provides an in-depth analysis of the advantages and disadvantages of the different architectures, enabling other researchers to identify a setting that best suits their requirements.

D. Limitations and Future Work

DSWx was used as the training data throughout this work. While it can sufficiently identify water and artifacts such as clouds and shadows, there are inherent limitations in the dataset that could affect our model's performance upon deployment. In particular, snow/ice labels can erroneously occur in areas near the equator and in low elevation areas due to coloration from high sediment concentrations or areas in the water where waves are breaking. Cloud labels from DSWx can also be dilated and, thus, cover areas that are not necessarily cloud. At the same time, there are instances where DSWx labels fail to detect clouds over water, which are mislabeled as nonwater.

Furthermore, we recognize that we are, in essence, making a model of a model by predicting DSWx output. Ideally, we would have a robust dataset of high quality labels generated from the ground, or from digitization of HLS data. In this way, we could assess our performance better—the best we can do is to recreate DSWx, so any errors there become our errors. Our use of the LANA dataset was, therefore, purposeful to assess model performance against manually digitized labels, even though the data volumes for LANA are far smaller. We are encouraged by our performance relative to LANA and use these results as strong evidence for our claims of skill for our multitask model.

Nonetheless, the framework we introduce in this work is broadly applicable for masking applications beyond learning from DSWx. Manual labels (e.g., cloud labels from the LANA dataset), when available, can also be used to fine tune the model. Once an adequate quantity of manually labeled data is available, the same model we introduced could be used, with similar performance as shown in our experiments in Table VI. This fine tuning, coupled with pretraining on large remote sensing datasets such as Satlas and using a sufficiently large architecture such as DeepLabv3+ or Swin-T could serve as a starting point for future research on satellite masking and reflectance-based estimation.

VI. Conclusion

Experiments in this article show that our proposed multitask models do well to supplement global surface water analysis, illustrating the simultaneous identification of different types of pixels (i.e., water, cloud, cloud shadow, terrain shadow, and snow/ice) in satellite images can result in more accurate masks with a faster runtime. We were able to speedup the runtime by as much as 30× while using less than half of the standard memory requirement. At the same time, we show that the introduction of our multitask model in a downstream application results in a better performance with an RMSE reduction of 2.64 mg/L for SSC estimation. In particular, the replacement of several modules with a single multitask model for isolating good quality water pixels results in more accurate SSC predictions.

While we show multiple options and comparisons across different architectures, we recommend future researchers to start with DeepLabv3+ (pretrained with ImageNet). Its performance on all masking tasks (water, cloud, cloud shadow, terrain shadow, snow/ice) indicate better performance than all baselines, and better than almost all the other multitask architectures. At the same time, its runtime and memory consumption are reasonable for its performance. From this starting point, we recommend future researchers to evaluate their needs and scale up or down as necessary (e.g., scale up to Swin-T for a better performance or scale down to MobileNetv3 for a smaller and faster model).

The framework presented here is an important step for global surface water analysis, and is part of the SWOT mission wrapper Confluence [77] used for SSC estimation. The proposed pipeline will be used to generate reliable, frequent, sediment flux estimations for every river in the SWORD database [78], based on global satellite observations. While we show results specific for water pixel identification and SSC estimation, our model can be applied to other areas as well that require distinguishing different types of entities in satellite images such as cloud and cloud shadows. The same proposed pipeline can also be used for other global downstream applications as a faster and resource-efficient alternative.

Acknowledgment

The authors would like to thank the Massachusetts Green High Performance Computing Center (MGHPCC) funded by the Massachusetts Technology Collaborative for the computational resources that supported this study.

References

[1] R.-R. Li and B.-C. Gao, “Remote sensing of suspended sediments and shallow coastal waters,” *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 3, pp. 559–566, Mar. 2003.

[2] J. C. Tilton, W. T. Lawrence, and A. J. Plaza, “Utilizing hierarchical segmentation to generate water and snow masks to facilitate monitoring change with remotely sensed image data,” *GIScience Remote Sens.*, vol. 43, no. 1, pp. 39–66, 2006.

[3] C. Huang et al., “Automated masking of cloud and cloud shadow for forest change analysis using Landsat images,” *Int. J. Remote Sens.*, vol. 31, no. 20, pp. 5449–5464, 2010.

[4] R. A. Aravena, M. B. Lyons, and D. A. Keith, “High resolution forest masking for seasonal monitoring with a regionalized and colourimetrically assisted chorologic typology,” *Remote Sens.*, vol. 15, no. 14, p. 3457, 2023.

[5] C. Atzberger, “Advances in remote sensing of agriculture: Context description, existing operational monitoring systems and major information needs,” *Remote Sens.*, vol. 5, no. 2, pp. 949–981, 2013.

[6] S. Valero et al., “Production of a dynamic cropland mask by processing remote sensing image series at high temporal and spatial resolutions,” *Remote Sens.*, vol. 8, no. 1, p. 55, 2016.

[7] M. Huang, N. Chen, W. Du, Z. Chen, and J. Gong, “DMBLC: An indirect urban impervious surface area extraction approach by detecting and masking background land cover on Google Earth image,” *Remote Sens.*, vol. 10, no. 5, p. 766, 2018.

[8] Q. Xiao, S. Ustin, and E. McPherson, “Using aviris data and multiple-masking techniques to map urban forest tree species,” *Int. J. Remote Sens.*, vol. 25, no. 24, pp. 5637–5654, 2004.

[9] H. K. Zhang, D. Luo, and D. P. Roy, “Improved landsat operational land imager (OLI) cloud and shadow detection with the learning attention network algorithm (LANA),” *Remote Sens.*, vol. 16, no. 8, p. 1321, 2024.

[10] S. Qiu, Z. Zhu, and B. He, “Fmask 4.0: Improved cloud and cloud shadow detection in landsats 4–8 and sentinel-2 imagery,” *Remote Sens. Environ.*, vol. 231, 2019, Art. no. 111205.

[11] Z. Zhu, S. Wang, and C. E. Woodcock, “Improvement and expansion of the Fmask algorithm: Cloud, cloud shadow, and snow detection for Landsats 4–7, 8, and sentinel 2 images,” *Remote Sens. Environ.*, vol. 159, pp. 269–277, 2015.

[12] X. Chen, L. Liu, X. Zhang, S. Xie, and L. Lei, “A novel water change tracking algorithm for dynamic mapping of inland water using time-series remote sensing imagery,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 1661–1674, 2020.

[13] X. Yang, T. M. Pavelsky, L. P. Bendezu, and S. Zhang, “Simple method to extract lake ice condition from landsat images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 4202010.

[14] J. R. Gardner, X. Yang, S. N. Topp, M. R. Ross, E. H. Altenau, and T. M. Pavelsky, “The color of rivers,” *Geophysical Res. Lett.*, vol. 48, no. 1, 2021, Art. no. e2020GL088946.

[15] E. Dethier, C. Renshaw, and F. Magilligan, “Toward improved accuracy of remote sensing approaches for quantifying suspended sediment: Implications for suspended-sediment monitoring,” *J. Geophys. Res. Earth Surf.*, vol. 125, no. 7, 2020, Art. no. e2019JF005033.

[16] J. Gardner, T. Pavelsky, S. Topp, X. Yang, M. R. Ross, and S. Cohen, “Human activities change suspended sediment concentration along rivers,” *Environ. Res. Lett.*, vol. 18, no. 6, 2023, Art. no. 064032.

[17] M. Zhang, Q. Dong, T. Cui, C. Xue, and S. Zhang, "Suspended sediment monitoring and assessment for yellow river estuary from landsat TM and ETM imagery," *Remote Sens. Environ.*, vol. 146, pp. 136–147, 2014.

[18] T. Langhorst, T. Pavelsky, M. Harlan, E. Friedmann, and C. J. Gleason, "Simultaneous remote sensing of river discharge and suspended sediment on the Sagavanirktok river, Alaska," in *Proc. AGU Fall Meeting Abstr.*, 2022, vol. 2022, pp. H32G–0 6.

[19] A. Narayanan, S. Cohen, and J. R. Gardner, "Riverine sediment response to deforestation in the Amazon basin," *Earth Surf. Dyn.*, vol. 12, no. 2, pp. 581–599, 2024.

[20] H. Xu, "Modification of normalised difference water index (NDWI) to enhance open water features in remotely sensed imagery," *Int. J. Remote Sens.*, vol. 27, no. 14, pp. 3025–3033, 2006.

[21] Y. Du, Y. Zhang, F. Ling, Q. Wang, W. Li, and X. Li, "Water bodies' mapping from sentinel-2 imagery with modified normalized difference water index at 10-m spatial resolution produced by sharpening the SWIR band," *Remote Sens.*, vol. 8, no. 4, p. 354, 2016.

[22] G. Donchyts, J. Schellekens, H. Winsemius, E. Eisemann, and N. Van de Giesen, "A 30 m resolution surface water mask including estimation of positional and thematic differences using landsat 8, SRTM and openstreetmap: A case study in the murray-darling basin, Australia," *Remote Sens.*, vol. 8, no. 5, p. 386, 2016.

[23] V. V. Salomonson and I. Appel, "Estimating fractional snow cover from modis using the normalized difference snow index," *Remote Sens. Environ.*, vol. 89, no. 3, pp. 351–360, 2004.

[24] E. Greenberg et al., "An improved scheme for correcting remote spectral surface reflectance simultaneously for terrestrial BRDF and water-surface sunglint in coastal environments," *J. Geophys. Res. Biogeosci.*, vol. 127, no. 3, 2022, Art. no. e2021JG006712.

[25] B.-C. Gao and R.-R. Li, "Correction of sunglint effects in high spatial resolution hyperspectral imagery using SWIR or NIR bands and taking account of spectral variation of refractive index of water," *Adv. Environ. Eng. Res.*, vol. 2, no. 3, pp. 1–15, 2021.

[26] D. Yamazaki et al., "A high-accuracy map of global terrain elevations," *Geophysical Res. Lett.*, vol. 44, no. 11, pp. 5844–5853, 2017.

[27] J. Zhang et al., "Delivering arbitrary-modal semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 1136–1147.

[28] M. Xu, Z. Zhang, F. Wei, H. Hu, and X. Bai, "Side adapter network for open-vocabulary semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 2945–2954.

[29] F. Isikdogan, A. C. Bovik, and P. Passalacqua, "Surface water mapping by deep learning," *IEEE J. Sel. topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 11, pp. 4909–4918, Nov. 2017.

[30] D. Chai, S. Newsam, H. K. Zhang, Y. Qiu, and J. Huang, "Cloud and cloud shadow detection in landsat imagery based on deep convolutional neural networks," *Remote Sens. Environ.*, vol. 225, pp. 307–316, 2019.

[31] A. Vaswani et al., "Attention is all you need," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[32] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 801–818.

[33] A. Howard et al., "Searching for mobilenetv3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1314–1324.

[34] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[35] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Representations*, 2021.

[36] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10012–10022.

[37] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 843–852.

[38] C. Schuhmann et al., "LAION-5B: An open large-scale dataset for training next generation image-text models," *Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 25278–25294, 2022.

[39] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[40] R. Girshick, "Fast r-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.

[41] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.

[42] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2961–2969.

[43] J. Jakubik et al., "Foundation models for generalist geospatial artificial intelligence," Oct. 2023, *arxiv:2310.18660*.

[44] F. Bastani, P. Wolters, R. Gupta, J. Ferdinando, and A. Kembhavi, "Satlaspretrain: A large-scale dataset for remote sensing image understanding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 16772–16782.

[45] J. W. Jones, "Improved automated detection of subpixel-scale inundation—revised dynamic surface water extent (DSWE) partial surface water tests," *Remote Sens.*, vol. 11, no. 4, p. 374, 2019.

[46] M. Claverie et al., "The Harmonized Landsat and Sentinel-2 surface reflectance data set," *Remote Sens. Environ.*, vol. 219, pp. 145–161, 2018.

[47] M. G. Bato et al., "A first look at the opera surface water extent and land surface disturbance products and their applications," in *Proc. EGU Gen. Assem. Conf. Abstr.*, 2023, pp. EGU–10200.

[48] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, pp. 41–75, 1997.

[49] Y. Zhang, Y. Wei, and Q. Yang, "Learning to multitask," *Adv. Neural Inf. Process. Syst.*, vol. 31, 2018.

[50] L. F. Isikdogan, A. Bovik, and P. Passalacqua, "Seeing through the clouds with deepwatermap," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 10, pp. 1662–1666, Oct. 2020.

[51] E. Dethier, "Suspended sediment concentration database, hydroshare," 2019. [Online]. Available: http://www.hydroshare.org/resource/2ee7d421618a4873b9906540d047ced4

[52] C. Färber et al., "Water quality at the global scale: Gemstat database and information system," in *Proc. EGU Gen. Assem. Conf. Abstr.*, 2018, p. 15984.

[53] J. Hartmann, R. Lauerwald, and N. Moosdorf, "A brief overview of the global river chemistry database, glorich," *Procedia Earth Planet. Sci.*, vol. 10, pp. 23–27, 2014.

[54] L. V. Lucchese et al., "Modeling suspended sediment concentration using artificial neural networks, an effort towards global sediment flux observations in rivers from space," in *Proc. Copernicus Meetings*, 2024, p. 6548.

[55] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[56] B. Baheti, S. Innani, S. Gajre, and S. Talbar, "Semantic scene segmentation in unstructured environment with modified DeepLabV3+," *Pattern Recognit. Lett.*, vol. 138, pp. 223–229, 2020.

[57] T. Guo, J. Dong, H. Li, and Y. Gao, "Simple convolutional neural network on image classification," in *Proc. IEEE 2nd Int. Conf. Big Data Anal.*, 2017, pp. 721–724.

[58] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," in *Proc. Int. Conf. Learn. Representations*, 2021.

[59] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 807–814.

[60] A. L. Maas et al., "Rectifier nonlinearities improve neural network acoustic models," in *Proc. Int. Conf. Mach. Learn.*, Atlanta, GA, 2013, vol. 30, no. 1, p. 3.

[61] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Med. Image Comput. Comput.-Assisted Interv. 18th Int. Conf.*, 2015, pp. 234–241.

[62] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[63] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*.

[64] J. Donahue et al., "DECAF: A deep convolutional activation feature for generic visual recognition," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 647–655.

[65] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3150–3158.

[66] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 580–587.

[67] M. Wieland, Y. Li, and S. Martinis, "Multi-sensor cloud and cloud shadow segmentation with a convolutional neural network," *Remote Sens. Environ.*, vol. 230, 2019, Art. no. 111203.

[68] N. Otsu et al., "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285–296, pp. 23–27, 1975.

[69] J. Soimasuo, H. Cho, and M. Neteler, "Sunmask algorithm." 2001. [Online]. Available: https://grass.osgeo.org/grass83/manuals/r.sunmask.html

[70] J. J. Michalsky, "The astronomical almanac's algorithm for approximate solar position (1950–2050)," *Sol. Energy*, vol. 40, no. 3, pp. 227–235, 1988.

[71] J. Spencer, "Fourier series representation of the position of the sun," *Search*, vol. 2, no. 5, p. 172, 1971.

[72] J. C. Zimmerman, "Sun-pointing programs and their accuracy," Sandia National Lab. (SNL-NM), Albuquerque, NM, USA, Tech. Rep. SAND-81-0761, 1981.

[73] M. Minsky and S. Papert, "An introduction to computational geometry," *Cambridge Tiass.*, HIT, vol. 479, no. 480, p. 104, 1969.

[74] W. Wang and Z. Shi, "An all-scale feature fusion network with boundary point prediction for cloud detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2021, Art. no. 8020705.

[75] K. Hu, D. Zhang, and M. Xia, "CDUNet: Cloud detection UNet for remote sensing imagery," *Remote Sens.*, vol. 13, no. 22, p. 4533, 2021.

[76] X. Yao, Q. Guo, and A. Li, "Cloud detection in optical remote sensing images with deep semi-supervised and active learning," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, 2023, Art. no. 6006805.

[77] N. Tebaldi et al., "Confluence: An open-source framework for swot discharge estimation and value-added products," in *Proc. AGU Fall Meeting Abstr.*, 2021, vol. 2021, pp. H45O–1346.

[78] E. H. Altenau, T. M. Pavelsky, M. T. Durand, X. Yang, R. P. d. M. Frasson, and L. Bendezu, "The surface water and ocean topography (SWOT) mission river database (SWORD): A global river network for satellite data products," *Water Resour. Res.*, vol. 57, no. 7, 2021, Art. no. e2021WR030054.

Rangel Daroya received the B.S. degree in electronics and communications engineering and the M.S. degree in electrical engineering from the University of the Philippines, Quezon City, Philippines, in 2017 and 2020, respectively. She is currently working toward the doctoral degree in computer science with the College of Information and Computer Sciences, University of Massachusetts, Amherst, MA, USA.

Her research interests include computer vision, explainable AI, and remote sensing.

Luisa Vieira Lucchese received the B.S. degree in civil engineering and the M.S. and Ph.D. degrees in water resources and environmental sanitation from Universidade Federal do Rio Grande do Sul, Porto Alegre, Rio Grande do Sul, Brazil, in 2015, 2018, and 2022, respectively.

She is currently a Research Assistant Professor with the Department of Geology and Environmental Science, University of Pittsburgh, Pittsburgh, PA, USA. Her research interests include remote sensing, hydrology, natural hazards, explainable AI, and spatial data science.

Travis Simmons received the B.S. degree in biology, minors in environmental science and data analytics from the College of Coastal Georgia, Brunswick, GA, USA, in 2022.

He is currently a Research Fellow with the Department of Civil and Environmental Engineering, University of Massachusetts, Amherst, MA, USA. His research interests include hydrology, data engineering, and pipeline development.

Punwath Prum received the B.S. degree in land management and land administration from the Royal University of Agriculture, Phnom Penh, Cambodia, in 2015 and the M.S. degree in environmental observation and informatics from the University of Wisconsin, Madison, WI, USA, in 2020. He is currently working toward the doctoral degree in geology and environmental science with the Department of Geology and Environmental Science, University of Pittsburgh, Pittsburgh, PA, USA.

His research interests include studying the impacts of human intervention and climate on water resources using remote sensing, modeling, and field observation.

Tamlin Pavelsky received the B.A. degree in geography from the Department of Geography, Middlebury College, Middlebury, VT, USA, in 2001 and the Ph.D. degree in geography from the University of California, Los Angeles, Los Angeles, CA, USA, in 2008.

He is currently a Professor with the Department of Earth, Marine and Environmental Sciences, University of North Carolina, Chapel Hill, NC, USA. He is also the U.S. Hydrology Science Lead for the NASA Surface Water and Ocean Topography (SWOT) Satellite Mission. His research interests are focused on the intersections between hydrology, satellite remote sensing, and climate change.

John Gardner received the B.S./B.A. degree in environmental science and geography from the University of Missouri, Columbia, MO, USA in 2010, the M.S. degree in marine, estuarine, and environmental science from University of Maryland, College Park, MD, USA, in 2014, and the Ph.D. degree in environmental science from Duke University, Durham, NC, USA, in 2018.

He is currently an Assistant Professor with the Department of Geology and Environmental Science, University of Pittsburgh, Pittsburgh, PA, USA. He is also the Associate Director with the Pittsburgh Water Collaboratory, Pittsburgh, PA, USA. His research interests focus on how rivers, lakes, and their landscapes move water, sediment, and elements across continents.

Colin J. Gleason received the B.S. degree in forest engineering and the M.S. degree in geospatial engineering from SUNY Environmental Science and Forestry in Syracuse, Syracuse, NY, USA, in 2009 and 2011, respectively, and the Ph.D. degree in geography with University of California Los Angeles, Los Angeles, CA, USA, in 2016.

He is currently the Armstrong Professional Development Professor with the University of Massachusetts, Amherst, Amherst, MA, USA. He was an Assistant Professor with the same institution. He is also a global hydrologist and geomorphologist that uses primarily satellite data to ask and answer questions about rivers.

Dr. Gleason is a member of American Geophysical Union (AGU) and European Geosciences Union (EGU).

Subhransu Maji received the B.Tech. degree in computer science and engineering from the Indian Institute of Technology Kanpur, Kanpur, India, in 2006 and the Ph.D. degree in computer science from the University of California, Berkeley, CA, USA, in 2011.

He is currently an Associate Professor with the College of Information and Computer Sciences, University of Massachusetts, Amherst, Amherst, MA, USA. He was a Research Assistant Professor with Toyota Technological Institute at Chicago, Chicago, IL, USA. His research interests include computer vision and machine learning.

Dr. Maji was the recipient of a Google graduate fellowship, NSF CAREER Award (2018), and a best paper honorable mention at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2018. He serves on the editorial board of the *International Journal of Computer Vision* (IJCV).