remote sensing

MDPI

Article

A Self-Trained Model for Cloud, Shadow and Snow Detection in Sentinel-2 Images of Snow- and Ice-Covered Regions

Kamal Gopikrishnan Nambiar [1], Veniamin I. Morgenshtern [1,*], Philipp Hochreuther [2], Thorsten Seehaus [2] and Matthias Holger Braun [2]

[1] Chair of Multimedia Communications and Signal Processing, Friedrich-Alexander-Universität Erlangen-Nürnberg, 91054 Erlangen, Germany; kamal.nambiar@fau.de
[2] Institute of Geography, Friedrich-Alexander-Universität Erlangen-Nürnberg, 91054 Erlangen, Germany; philipp.hochreuther@fau.de (P.H.); thorsten.seehaus@fau.de (T.S.); matthias.h.braun@fau.de (M.H.B.)
* Correspondence: veniamin.morgenshtern@fau.de

Abstract: Screening clouds, shadows, and snow is a critical pre-processing step in many remote-sensing data processing pipelines that operate on satellite image data from polar and high mountain regions. We observe that the results of the state-of-the-art Fmask algorithm are not very accurate in polar and high mountain regions. Given the unavailability of large, labeled Sentinel-2 training datasets, we present a multi-stage self-training approach that trains a model to perform semantic segmentation on Sentinel-2 L1C images using the noisy Fmask labels for training and a small human-labeled dataset for validation. At each stage of the proposed iterative framework, we use a larger network architecture in comparison to the previous stage and train a new model. The trained model at each stage is then used to generate new training labels for a bigger dataset, which are used for training the model in the next stage. We select the best model during training in each stage by evaluating the multi-class segmentation metric, mean Intersection over Union (mIoU), on the small human-labeled validation dataset. This effectively helps to correct the noisy labels. Our model achieved an overall accuracy of 93% compared to the Fmask 4 and Sen2Cor 2.8, which achieved 75% and 76%, respectively. We believe our approach can also be adapted for other remote-sensing applications for training deep-learning models with imprecise labels.

Keywords: deep learning; semi-supervised learning; semantic segmentation; self-training; automatic cloud screening; Fmask; Sentinel-2 imagery

Citation: Nambiar, K.G.; Morgenshtern, V.I.; Hochreuther, P.; Seehaus, T.; Braun, M.H. A Self-Trained Model for Cloud, Shadow and Snow Detection in Sentinel-2 Images of Snow- and Ice-Covered Regions. *Remote Sens.* **2022**, *14*, 1825. https://doi.org/10.3390/rs14081825

Academic Editors: Annett Bartsch, Gareth Rees and Neil Arnold

Received: 26 February 2022
Accepted: 5 April 2022
Published: 10 April 2022

1. Introduction

Satellite imagery provides a vital source of information for a wide range of remote-sensing applications. However, such imagery is often contaminated with cloud and cloud shadows. Screening cloud and shadows is a critical preprocessing step that needs to be undertaken before any meaningful analysis can be performed on these images.

Traditional algorithms use threshold functions on spectral values of the image to generate cloud and shadow masks. In addition to this, metadata, such as solar zenith and azimuth angles, and elevation, are also used in computing these masks. The Fmask algorithm is widely accepted in the remote sensing community as a reliable method for generating cloud and cloud shadow masks [1]. Though the initial version of the Fmask was developed for Landsat-5 and Landsat-7 imagery, subsequent publications adapted the algorithm for use in Landsat-8 and Sentinel-2 imagery and proposed further improvements [2,3]. CFmask, a derivative of the Fmask algorithm, is used by the U.S. Geological Survey (USGS) in their production environment. The Sen2Cor process used by the European Space Agency (ESA) to process its Sentinel-2 products generates a scene classification map that detects clouds, shadows, water and snow [4]. However, the results of these algorithms are known to be erroneous when the underlying surface is brightly colored, for example, in the case of

ice and snow, white building tops, salt lakes, etc. The detection of shadows is not reliable on darker surfaces. Water bodies are often misclassified as shadows and vice versa. In addition to Fmask and Sen2Cor, several texture-based and threshold-based approaches have been proposed [5–7]. Early attempts at using data-driven methods, such as support vector machines [8,9], random forest [10,11] and Markov random fields [12,13], for this task, have shown improvements over the traditional rule-based methods. In this paper, we propose a deep-learning-based model that specializes in cloud detection in polar and high mountain regions that are snow- and ice-covered for most parts of the year.

Over the past decade, deep convolutional neural networks (DCNNs) have made giant leaps in the field of computer vision and have found application in domains that require very high precision, such as medical imaging and autonomous driving [14,15]. In contrast to single pixel classification methods, DCNNs learn spatial relationships between the neighboring pixels in the image. Hence, the network can make use of texture and structural information, in addition to the spectral properties, to distinguish objects in the scene—this contributes towards improved detection performance. DCNNs are typically trained using large amounts of correctly labeled data. However, labeled data is very expensive because it is time-consuming to produce.

The open access policy of the Copernicus programme has led to an abundant use of optical Sentinel-2 data. No cloud mask is provided with this optical data that reliably detects clouds over bright targets or shadows over dark targets. While there are a few datasets for cloud detection, they were unsuitable for direct use in our multi-class segmentation problem. In this study, we propose a self-training framework in order to train our model. The trained model in our work segments a given Sentinel-2 L1C scene into six classes: *No-Data*, *Clear-Sky Land*, *Cloud*, *Shadow*, *Snow* and *Water*. We use a large dataset with labels generated using the Fmask algorithm for the training, and a small human-labeled dataset for validation. The validation dataset contains numerous examples where the Fmask classification has given incorrect labels.

The trained model, when compared with widely used automatic methods, such as Fmask 4 and Sen2Cor 2.8, on our test dataset, showed significant improvement in various segmentation metrics. Interestingly, the model achieved better performance than its teacher, Fmask, that was used to automatically generate the labels for training.

To facilitate benchmarking and further research, we have made the labeled dataset openly accessible via the Pangaea data center (https://doi.org/10.1594/PANGAEA.942321, accessed on 25 February 2022); the code is available in the GitHub repository (https://github.com/kmlnbr/deep-fmask, accessed on 25 February 2022).

2. Previous Related Work

2.1. Self-Training

Given the scarcity of large, labeled datasets required to train the increasingly complex neural network models, self-training has recently attracted attention in the research community [16–18]. Self-training refers to a learning paradigm where a base model, which was trained using a smaller labeled dataset, is used to generate the pseudo-labels for the unlabeled data that will be used for training another model [19]. Xie et al. proposed a self-training method that achieved a 2% improvement over the state-of-the-art model on an Imagenet classification task using weakly labeled images [18]. Babakhin et al. used a similar iterative training method for segmenting salt deposits in seismic images [20] and Chen et al. applied a similar iterative training method for scene segmentation from video sequences [21]. In each of these papers, small clean labeled datasets were used in combination with a large unlabeled dataset to train the model. In our research, we used a labeled dataset generated using the Fmask algorithm and a large unlabeled dataset for training. Unlike the previous papers, the Fmask algorithm did not always provide clean labels. Hence, we used different regularization techniques in our training to ensure that the model was robust to label noise in the training dataset. Yilmaz and Heckel showed that deep neural networks, trained using an early stopping strategy, fit clean labels faster

than noisy labels [22]. Experiments performed on 10%- to 50%-correctly labeled training datasets have produced higher test performance than standard training on a clean dataset for the Imagenet and CIFAR-10 classification tasks.

2.2. Cloud Detection Algorithms

Early cloud-detection algorithms relied heavily on feature engineering to produce accurate cloud detection models [8,23,24]. The Landsat 8 Cloud Cover Assessment (CCA) dataset, designed to test the effectiveness of cloud detection algorithms, has helped in the development of new deep-learning-based methods for Landsat imagery [25–27]. Scene classification using deep neural networks is treated as a semantic segmentation task, and several variants of the well-established encoder-decoder architectures, U-Net and Fully Convolutional Network (FCN), have been proposed. Some of the pioneering studies in this domain for Landsat-8 imagery include RS-Net [28], Cloud-Net [29], and MF-CNN [30]. Similarly, for images from the GaoFen satellite series operated by the China National Space Administration (CNSA), Zhan et al. used an FCN with a multiscale prediction module to distinguish clouds and snow [31], and Yan et al. implemented a multilevel feature-fused segmentation network called MFFSNet, to perform cloud and cloud shadow detection [32]. Recently, several improvements, such as attention mechanisms [33,34], and novel convolution techniques [35], have contributed to advancing the state-of-the-art in this domain. In contrast to the methods discussed above, which rely on manually labeled, pixel-level annotations, Li et al. proposed a weakly supervised method, trained using block-level labels that indicate the presence or absence of clouds in a given block [36]. The network was used to generate cloud activation maps, which were subsequently segmented, using a statistical threshold based on clear-sky values, to obtain pixel-level cloud detection output.

Liu et al. proposed a neural network with a residual architecture called CloudNet to predict clouds and haze from Sentinel-2 imagery [37]. This method uses Sen2Cor corrected data from Taiwan for training and primarily functions as a cloud-detection algorithm. Li et al. used a lightweight network, trained using a manually labeled dataset over diverse land cover and climatic conditions in mainland China, for performing cloud detection [38]. In contrast to these methods, we use data from polar and high mountain regions and address a more challenging multi-class segmentation problem.

Hughes and Kennedy proposed a fully convolutional network that segmented Landsat-8 imagery, and Hollstein et al. used a decision-tree model that segmented Sentinel-2 imagery into the same classes as our investigation [10,39].

3. Materials and Methods

3.1. Dataset

The Sentinel-2 L1C data used for this study was downloaded from the Copernicus Open Access Data Hub. The data was captured by the Sentinel-2A and Sentinel-2B satellites using their onboard multi-spectral instrument (MSI) that captures images in 13 frequency bands. The description of the bands used in this study is provided in Table 1. The L1C data contains the top-of-atmosphere (TOA) reflectance values and each scene covers an area of 100 km $\times$ 100 km.

The sensing period for the images used in our study was between October 2019 and December 2020. We used 96 scenes in the training dataset, 22 scenes in the validation dataset and 23 scenes in the test dataset. The sites of these scenes were randomly selected in the polar and tundra regions. We also included three sites in mountainous terrain that were not in the polar and tundra region. The geographic distribution of the data is shown in Figure 1. The tile-wise quarterly distribution of the Sentinel-2 scenes for the training, validation and test datasets is provided in Tables A1, A2 and A3, respectively.

Table 1. Description of the spectral bands from Sentinel-2A [40] that were used as input in our study. The spectral bands of its twin satellite, Sentinel-2B, also had the same resolution for the respective bands and only varied in the central wavelength by a small fraction. Data from both Sentinel-2 satellites were used in our study without any distinction regarding their source.

Band No.	Central Wavelength (nm)	Resolution (m)	Image Size (Pixels)	Details
2	492.4	10	10,980 × 10,980	Blue Band
3	559.8	10	10,980 × 10,980	Green Band
4	664.6	10	10,980 × 10,980	Red Band
8	832.8	10	10,980 × 10,980	Near Infrared
11	1613.7	20	5490 × 5490	Short-wave Infrared
12	2202.4	20	5490 × 5490	Short-wave Infrared

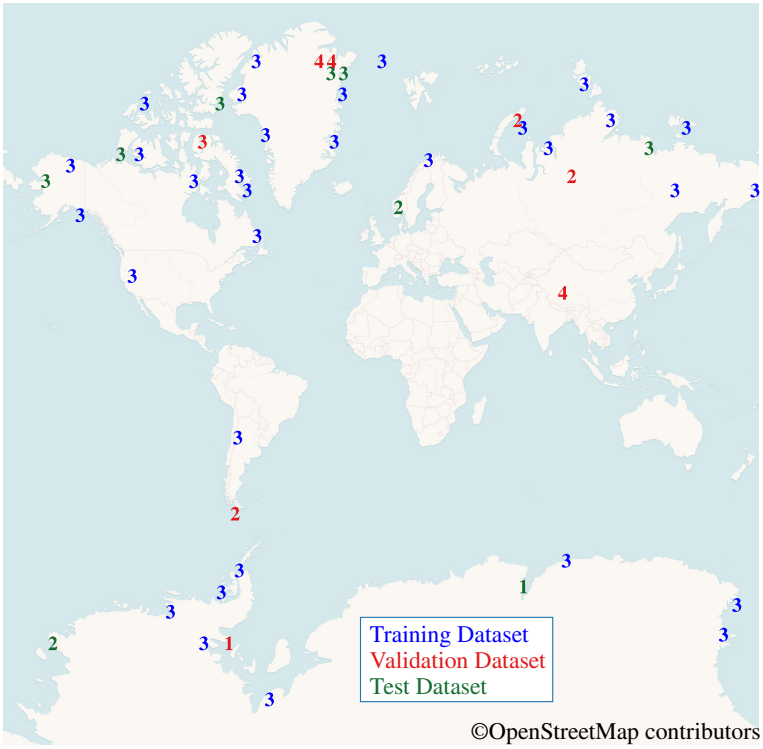

Figure 1. Geographic distribution of the datasets. The labeled numbers indicate the number of Sentinel-2 scenes used from a given site, and their colors indicate the dataset to which they belong.

The six classes that we used in our model: *No-Data*, *Clear-Sky Land*, *Cloud*, *Shadow*, *Snow* and *Water*, were the same class labels as were used in the segmentation output of the F-Mask algorithm. The *No-Data* class was used to indicate pixels in the image that were saturated or defective and regions along the edge of the image where sensor data was not available. Such pixels can be identified using the quality indicator information provided along with the Sentinel-2 imagery. Hence, we did not include this class in the labeled dataset. The *Clear-Sky Land* class was used for snow- and cloud-free land, which was usually rocky surfaces in the sites used for this study. The *Shadow* class was used for all types of shadow, irrespective of the object that cast the shadow, or the underlying surface on which the shadow fell. The samples in the validation set were selected so that many examples where the Fmask algorithm produced an incorrect label (as per our visual assessment) were present. The labeling for the validation and test dataset was performed manually using the QGIS software [41] package. In addition to the true color image, the short-wave infrared band (B11) and the near infrared band (B08) were used in our labeling process. Labeling of very small targets was performed occasionally using a thresholding operation in the local neighborhood of the target.

The Fmask labels were generated using Fmask 4.3. According to its authors, this version of Fmask offers substantially better cloud, cloud shadow, and snow detection results compared to the previous versions for Sentinel-2 imagery [2,3,42]. The cloud buffer distance, and the cloud shadow buffer distance were set to 0 in the Fmask configuration settings. All other configuration settings were set to their default values, and the Fmask output was computed at 20 m (5490 × 5490 pixels) resolution.

We also compared our model performance against the results from Sen2Cor 2.8. The Sen2cor algorithm is part of the European Space Agency's (ESA) processing pipeline, used for generating the L2A images [4]. This algorithm produces a scene classification (SCL) image, in addition to various quality indicators and bottom-of-atmosphere (BOA) images. The resolution of the SCL was set to 20 m and no additional digital elevation model (DEM) was provided. While the Fmask segmented the image into the same six classes that we used for training our model, Sen2Cor provided a more comprehensive class organization. For example, the Sen2Cor scene classification mask has four classes for clouds: cloud high probability, cloud medium probability, cloud low probability and thin cirrus clouds. To perform the comparison, we combined multiple classes from the Sen2Cor mask into the six classes used in our model, as shown in Table 2.

Table 2. Regrouping of Sen2Cor labels.

Sen2Cor Label	New Label
No-Data Saturated/Defective Pixels Unclassified	No-Data
Vegetation Non-vegatated	Clear-Sky Land
Cloud High Probability Cloud Medium Probability Cloud Low Probability Thin Cirrus Clouds	Cloud
Cloud Shadow Shadows/ Dark Area Pixels	Shadow
Water	Water
Snow	Snow

As shown in Table 1, the spectral bands in the Sentinel-2 L1C product are provided in different resolutions. We resampled all the bands to a resolution of 20 m using bicubic interpolation. Due to hardware limitations, it was not possible to train the network with the resampled image of size 5490 × 5490. Hence, we split the images into sub-scenes of size 254 × 254 pixels for training. Since the optical sensor data from the satellite may not always be aligned to fill the complete image, the edges of the image tend to have zero-valued pixels, i.e., no data. After splitting the image into sub-scenes, we discarded those sub-scenes in which all pixels were zero-valued. We applied zero-padding of one pixel around each sub-scene, which increased the network input size to 256 × 256. Since the network did not have any fully connected layers (which requires a fixed input size), we were able flexibly to use a larger sub-scene size for prediction. In the prediction step, we used overlapping sub-scenes of size 510 × 510, and we also applied zero-padding, as in the training step. Before stitching together the network output from each sub-scene, we clipped out a three-pixel border around the sub-scene to eliminate the uncertain network predictions from the zero-padding [39].

The input to the network consisted of seven channels: the Sentinel-2 RGB bands (B02, B03, B04), near infrared band (B08), the short-wave infrared bands (B11, B12) and the normalised difference snow index (NDSI). The NDSI is a well-known index that exploits

the difference in the spectral reflectance of the green band (B03) and shortwave infrared band (B11) to detect snow [43]. The NDSI is computed pixel-wise as follows:

$$\text{NDSI} = \frac{\text{B03} - \text{B11}}{\text{B03} + \text{B11}}. \quad (1)$$

3.2. Neural Network Architecture

The U-Net [14] and its variants are widely used neural network architectures for semantic segmentation. The network displayed in Figure 2 consists of an encoder-decoder architecture. Each encoder block operates at a spatial resolution twice that of its succeeding encoder but has half the number of convolution filters compared to its succeeding encoder.

At each encoder and decoder block, there are two sequences, comprising a convolution layer, a ReLU layer, and a batch-normalization layer connected in series, followed by a spatial dropout. The convolution layers used here had a kernel size of 3×3 and a stride of 1.

The output of the encoder takes two paths. In one path, the output of the spatial dropout is downsampled by a factor of two in the spatial dimensions using a max-pool layer, and is provided as input to the next encoder. The other path, referred as the skip connection, is connected to the decoder block that operates, at the same resolution, on the other side of the network.

At each decoder, two inputs are received and these two inputs have to be concatenated as a single input before applying the layers of the decoder. The input received from the preceding layer is upsampled using a 2×2 up-convolution operation with a stride of 2, which results in an increase in spatial dimensions by a factor of two, and a decrease in the number of feature maps by a factor of two. The upsampled output is then concatenated with the input received from the encoder of the corresponding step via the skip connection (denoted as a blue box in the decoder block in Figure 2). The decoder layers are then applied to the concatenated output and the result is provided as the input to the next decoder.

This sequence is repeated until the final decoder that operates at the same spatial resolution as the input. At the final decoder, an additional 1×1 convolution, and a softmax layer, is applied to obtain the class probabilities for each pixel of the image. The network output has the same spatial dimensions as the input and has six channels, where each channel contains the probability map for the respective class. The segmentation map can be generated by selecting the class with the highest probability at each pixel position.

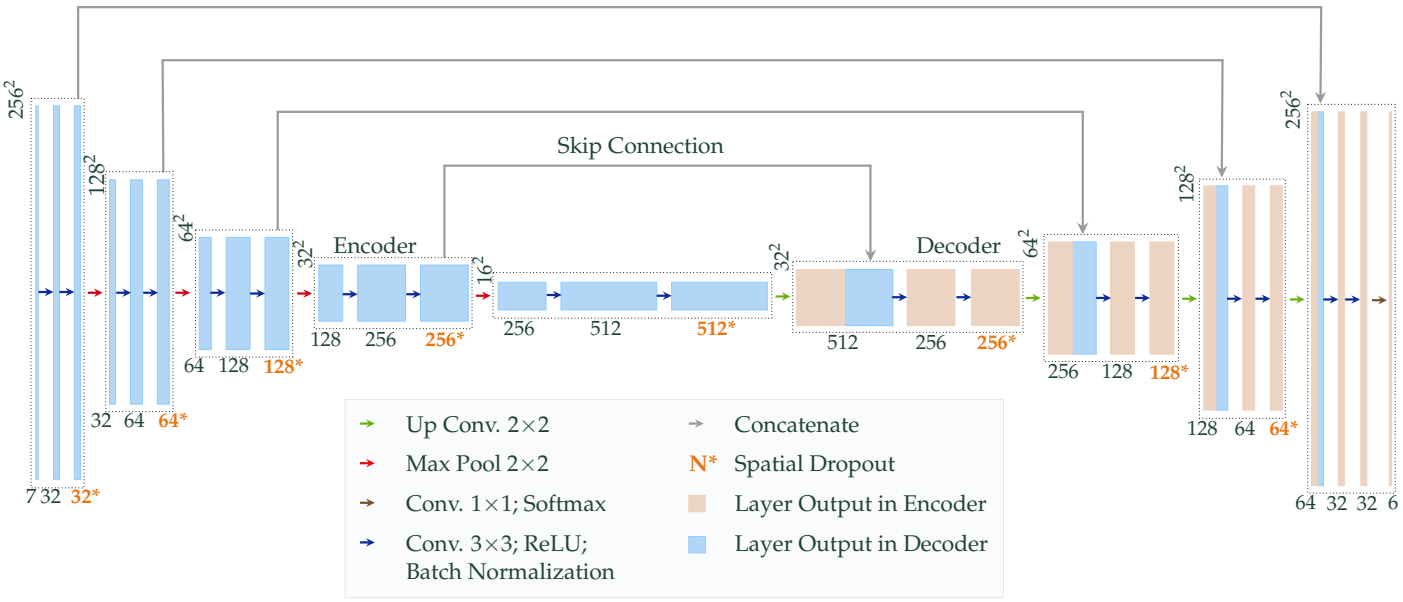

Figure 2. The U-Net architecture [14] with 32 start filters and a depth of 5, used in stage 2 of the self-training framework. The layers that constitute each encoder and decoder block are shown inside the boxes with dotted borders. The number of feature maps is indicated below each colored box. The resolution is the same for all layers in the encoder/decoder block; this is indicated at the top left of the dotted box. The output of the encoder, which is provided via the skip connection, is concatenated with the output of the up-convolution operation from the previous layer.

3.3. Self-Training Framework

We used a four-stage iterative approach to train our model. The network architecture and training data organization of the iterative approach is summarized in Table 3. The training dataset, consisting of sub-scenes of size 254 × 254 pixels, was divided randomly into four equal batches. In the first stage, a deep neural network with a modified U-Net architecture was trained using batch-1 of the training dataset. The *imperfect* labels used in this stage were generated automatically using the Fmask algorithm. We evaluated the model performance on the validation dataset with human-labeled data after each training epoch, and, importantly, the human-labeled data was never used as training data. Based on the performance metrics, we selected the model from the best epoch, and called the selected model the *teacher model*. Using the teacher model, we inferred labels for the data in batch-2 of the training dataset. We used only the labels for which the network confidence was greater than a threshold of 0.33 (twice the probability of a random guess). We assigned the label as 0 (*No-Data* label) to pixels where the confidence was lower than this threshold. In the second stage, we trained another neural network model, called the *student model*, using a combined dataset of batch-1 and batch-2. For the batch-1 data, we continued to use the Fmask labels exactly as in the previous stage. Once we had selected the best student model from stage 2 using the validation dataset, we made this student model our new teacher model. Using the new teacher model, we inferred new labels for batch-2 and batch-3 data. In each subsequent stage, we trained a new student model with a combined dataset of Fmask labels for batch-1, and model-inferred labels from the teacher model, i.e., the best model of the previous stage, for the remaining batches, as shown in Table 3.

A new batch of the data containing previously unseen images was available for training the model at each stage. Hence, the size of the training data increased linearly at each stage. A smaller model was used for the training in the first stage, and the model size incrementally increased at each stage with increase in the training data. This was done to make sure that the model did not overfit on the smaller training data in the early stages. Hence, we defined two hyperparameters, *number of start filters* and *depth*, which we used to control the model size. The number of start filters was the number of filters used in the

convolution layer present in the first encoder of the network. The number of filters in the subsequent encoders was defined as multiples of this parameter. The depth of the network set the number of encoder blocks used in the network. A network architecture with 32 start filters and a depth of 5 is illustrated in Figure 2.

Table 3. Network architecture and training data used at each stage of the iterative approach. The number of parameters in the network at each stage was controlled using the number of start filters and depth hyperparameters. The size of the training dataset given below was the number of 254 × 254 image patches.

	Network Architecture			Training Data		
Training Stage	**No. Start Filters**	**Depth**	**No. Parameters**	**Size**	**Training Batch**	**Label Source**
1	16	5	1.9 M	11,263	Batch-1	Fmask
2	32	5	7.8 M	22,524	Batch-1 Batch-2	Fmask Model Stage-1
3	24	6	17.5 M	33,785	Batch-1 Batch-2 Batch-3	Fmask Model Stage-2 Model Stage-2
4	32	6	31.1 M	45,046	Batch-1 Batch-2 Batch-3 Batch-4	Fmask Model Stage-3 Model Stage-3 Model Stage-3

3.4. Training Implementation

3.4.1. Regularization Techniques

We employed the regularization techniques described below at each stage of the self-training framework to aid in generalization of the trained model.

We used an online augmentation strategy where we performed augmentation "on the fly" before the data was provided as input to the network. The training dataset already consisted of a large number of data samples, and using the online strategy helped us achieve diversity across each training epoch without an explosive increase in the dataset size. The augmentations used included cutout [44], horizontal flip, vertical flip, and rotation in steps of 90°. These transformations were applied on randomly selected input samples and more than one transformation could be applied to the same image.

Unlike for fully connected layers, it has been shown that element-wise dropout is not very effective in the case of convolutional layers. The spatial dropout used in this study set entire feature maps of the output to zero instead of just individual pixels [45]. The application of spatial dropout leads to learning from an ensemble of sub-networks consisting of randomly selected feature maps and hence helps in regularizing the trained model.

Each stage of the self-training framework was trained for 100 epochs. An early stopping strategy that stopped the training if the average class accuracy on the validation dataset did not improve after 10 epochs, was employed. The Adam optimizer was used for the training with a weight decay of 1×10^{-5}. The weight decay resulted in ℓ_2 norm regularization for the network weights.

3.4.2. Loss Function

As shown in Table 4, the training data had a class imbalance problem. This could have resulted in the network predicting the abundant class more often than the classes that were scarce. Hence, we used a weighted cross-entropy function as the loss function for the training. The weights for each class were calculated using the median frequency balancing (MFB) method [46]. It has been shown that this technique is effective in semantic segmentation of small targets in remote-sensing images [47]. The weights were calculated using the frequency of the class in the training dataset: Let the set $C = \{1, 2, \ldots, M\}$ denote

the set of the M classes in the given segmentation problem. The frequency of class i, denoted as f_i, is the ratio of the number of pixels that are labeled as class i to the total number of pixels. The median of the class frequencies in the set C is given by $\text{median}(\{f_i | i \in C\})$. The weight w_i for class i is given by:

$$w_i = \frac{\text{median}(\{f_i | i \in C\})}{f_i}. \tag{2}$$

The target label used for training has a one-hot encoding, i.e., it is a vector whose elements are all zero except for one element corresponding to the true class set to one. The weighted cross entropy loss is given by:

$$\mathcal{L} = -\frac{1}{N} \sum_{j=1}^{N} \sum_{i=1}^{M} w_i \times y_i^{(j)} \times \log(\hat{y}_i^{(j)}), \tag{3}$$

where N is the number of training samples (pixels), w_i is the weight of class i calculated using the MFB method, $y_i^{(j)}$ is the target label at the dimension corresponding to class i for the training sample j, and $\hat{y}_i^{(j)}$ is the softmax output of the model at the dimension corresponding to class i.

Table 4. Class distribution of the datasets used for training and validation of the model. The size of each dataset is quantified as the number of 254 × 254 image patches.

			Class Distribution (%)					
Dataset	**Label Source**	**Size**	**No-Data**	**Clear-Sky Land**	**Cloud**	**Shadow**	**Snow**	**Water**
Train	Fmask	45,046	0.48	13.04	39.62	5.51	29.97	11.36
Validation	Human-labeled	∼204 [a]	0	10.64	40.70	24.80	13.83	10.03
Test	Human-labeled	∼180 [b]	0	5.59	55.71	12.07	18.28	8.34

[a] The labels in the validation dataset are of irregular shapes and of varying sizes. The validation dataset consists of 13,147,329 labeled pixels, which corresponds to approximately 204 images of size 254 × 254. [b] The test dataset consists of 11,596,941 labeled pixels, which corresponds to approximately 180 images of size 254 × 254.

3.5. Validation Metrics

The performance metrics used for the model evaluation were computed from a pixel-level confusion matrix: Consider the confusion matrix for an M-class segmentation problem shown in Figure 3. Each element in the matrix, denoted by $n_{i,j}$, is the total number of pixels that belong to class i and predicted as class j by the model. For class i,

$$\text{TP}_i = n_{i,i}, \ \text{FP}_i = \sum_{\substack{j=1,\\ j\neq i}}^{M} n_{j,i}, \text{ and } \text{FN}_i = \sum_{\substack{j=1,\\ j\neq i}}^{M} n_{i,j}, \tag{4}$$

are the number of true positives, false positives and false negatives, respectively. Precision or user accuracy for class i is defined as follows:

$$\mathcal{P}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i}, \tag{5}$$

i.e., the ratio of the number of pixels that were correctly predicted as class i to the total number of pixels that were predicted as class i. Recall or producer accuracy for class i is defined as follows:

$$\mathcal{R}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i}, \tag{6}$$

i.e., the ratio of the number of pixels that were correctly predicted as class i to the total number of pixels that actually belong to class i, i.e., the ground truth.

True Labels \ Model Labels	1	2	3	4	5	6
1	$n_{1,1}$	$n_{1,2}$	$n_{1,3}$	$n_{1,4}$	$n_{1,5}$	$n_{1,6}$
2	$n_{2,1}$	$n_{2,2}$	$n_{2,3}$	$n_{2,4}$	$n_{2,5}$	$n_{2,6}$
3	$n_{3,1}$	$n_{3,2}$	$n_{3,3}$	$n_{3,4}$	$n_{3,5}$	$n_{3,6}$
4	$n_{4,1}$	$n_{4,2}$	$n_{4,3}$	$n_{4,4}$	$n_{4,5}$	$n_{4,6}$
5	$n_{5,1}$	$n_{5,2}$	$n_{5,3}$	$n_{5,4}$	$n_{5,5}$	$n_{5,6}$
6	$n_{6,1}$	$n_{6,2}$	$n_{6,3}$	$n_{6,4}$	$n_{6,5}$	$n_{6,6}$

Figure 3. Confusion matrix of an M-class segmentation problem with $M = 6$. Each element in the matrix, denoted $n_{i,j}$, is the total number of pixels that belong to class i and predicted as class j by the model. For class 3, the True Positive pixels are represented by the green cell; the False Negative pixels are represented by the blue cells and the False Positive pixels are represented by the red cells.

Cases of over-segmentation and under-segmentation are often misinterpreted as good results when either recall or precision is analyzed individually. Hence, in semantic segmentation tasks, metrics such as the F1 Score and the Intersection over Union (IoU) are the preferred evaluation metrics. The F1 score is the harmonic mean of the precision and recall:

$$\text{F1 Score}_i = 2 \times \frac{\mathcal{R}_i \times \mathcal{P}_i}{\mathcal{R}_i + \mathcal{P}_i}. \tag{7}$$

Given the set of pixels that are predicted to be a particular class by the model, and the set of pixels that actually belong to that class (ground truth), IoU is defined as the ratio of the size of the intersection to the size of the union of these two sets. The IoU is expressed in terms of the number of true positives, true negatives and false positives as follows:

$$\text{IoU}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i + \text{FN}_i}. \tag{8}$$

The overall performance of the model can be evaluated using the mean Intersection over Union (mIoU) and the total accuracy. The mIoU for a segmentation task with M classes is defined as follows:

$$\text{mIoU} = \frac{1}{M} \sum_{i=1}^{M} \text{IoU}_i, \tag{9}$$

i.e., the mean of the IoU metric computed over each class. The total accuracy is then defined as follows:

$$\text{Total Accuracy} = \frac{\sum_{i=1}^{M} n_{i,i}}{\sum_{i=1}^{M} \sum_{j=1}^{M} n_{i,j}}, \tag{10}$$

i.e., the ratio of the number of correctly labeled pixels to the total number of pixels.

4. Results

We compared the performance of our model with two widely used methods for cloud masking, Fmask and Sen2Cor.

The confusion matrices comparing the prediction of Fmask, Sen2Cor and our model with the true labels in the test dataset are presented in Tables 5, 6 and 7, respectively. Table 8 shows that our model was able to achieve high precision and recall *simultaneously*. In the case of Fmask 4 and Sen2Cor 2.8, the high values of precision or recall for certain classes,

such as the class *Clear-Sky Land*, were accompanied by lower values of recall or precision, respectively. Therefore the overall segmentation performance suffered. In contrast, our model outperformed both these methods in this regard; this is clearly reflected in the F1 score and IoU metrics shown in Table 9 and in Figure 4. We also observed that the performance of the class *Water* was slightly better for Fmask compared to our model.

Table 5. Confusion matrix comparing the predictions of Fmask 4 with the true labels in the test dataset.

		Fmask 4 Labels						Recall
		No-Data	**Clear-Sky Land**	**Cloud**	**Shadow**	**Snow**	**Water**	
True Labels	**No-Data**	**0**	0	0	0	0	0	0.00
	Clear-Sky Land	0	**629,977**	7605	2727	3431	4772	0.97
	Cloud	0	647,645	**4,663,884**	98,931	1,050,628	82	0.72
	Shadow	73	67,800	32,191	**513,100**	585,632	201,325	0.37
	Snow	0	71	114,551	7800	**1,990,272**	7744	0.94
	Water	5	1123	1117	4823	420	**959,212**	0.99
Precision		0.00	0.47	0.97	0.82	0.55	0.82	

Table 6. Confusion matrix comparing the predictions of Sen2Cor 2.8 with the true labels in the test dataset.

		Sen2Cor 2.8 Labels						Recall
		No-Data	**Clear-Sky Land**	**Cloud**	**Shadow**	**Snow**	**Water**	
True Labels	**No-Data**	**0**	0	0	0	0	0	0.00
	Clear-Sky Land	164,756	**385,323**	85,479	10,415	870	1669	0.59
	Cloud	69,182	4772	**5,106,178**	2626	1,262,053	16,359	0.79
	Shadow	78,465	1529	32,579	**252,020**	333,368	702,160	0.18
	Snow	5238	0	88,040	8	**2,011,819**	15,333	0.95
	Water	221	0	511	2412	0	**963,556**	1.00
Precision		0.00	0.98	0.96	0.94	0.56	0.57	

Table 7. Confusion matrix comparing the predictions of our model with the true labels in the test dataset.

		Our Model Labels						Recall
		No-Data	**Clear-Sky Land**	**Cloud**	**Shadow**	**Snow**	**Water**	
True Labels	**No-Data**	**0**	0	0	0	0	0	0.00
	Clear-Sky Land	1	**573,931**	65,119	7777	1468	216	0.88
	Cloud	85	23,195	**6,111,259**	156,238	170,347	46	0.95
	Shadow	207	4232	10,126	**1,076,655**	39,761	269,140	0.77
	Snow	102	22	54,618	24,948	**2,032,404**	8344	0.96
	Water	0	0	0	5445	0	**961,255**	0.99
Precision		0.00	0.95	0.98	0.85	0.91	0.78	

Table 8. Comparison of precision and recall with Fmask 4 and Sen2Cor 2.8.

	Fmask 4		Sen2Cor 2.8		Our Model	
Class	**Precision**	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**
Clear-Sky Land	0.47	**0.97**	**0.98**	0.59	0.95	0.88
Cloud	0.97	0.72	0.96	0.79	**0.98**	**0.95**
Shadow	0.82	0.37	**0.94**	0.18	0.85	**0.77**
Snow	0.55	0.94	0.56	0.95	**0.91**	**0.96**
Water	**0.82**	0.99	0.57	**1.00**	0.78	0.99

Table 9. Comparison of F1 score and IoU with Fmask 4 and Sen2Cor 2.8.

	Fmask 4		Sen2Cor 2.8		Our Model	
Class	**F1 Score**	**IoU**	**F1 Score**	**IoU**	**F1 Score**	**IoU**
Clear-Sky Land	0.63	0.46	0.74	0.59	**0.92**	**0.85**
Cloud	0.83	0.70	0.87	0.77	**0.96**	**0.93**
Shadow	0.51	0.34	0.30	0.18	**0.81**	**0.68**
Snow	0.69	0.53	0.70	0.54	**0.93**	**0.87**
Water	**0.90**	**0.81**	0.72	0.57	0.87	0.77
Total Accuracy	0.76		0.75		**0.93**	
mIOU	0.57		0.53		**0.82**	

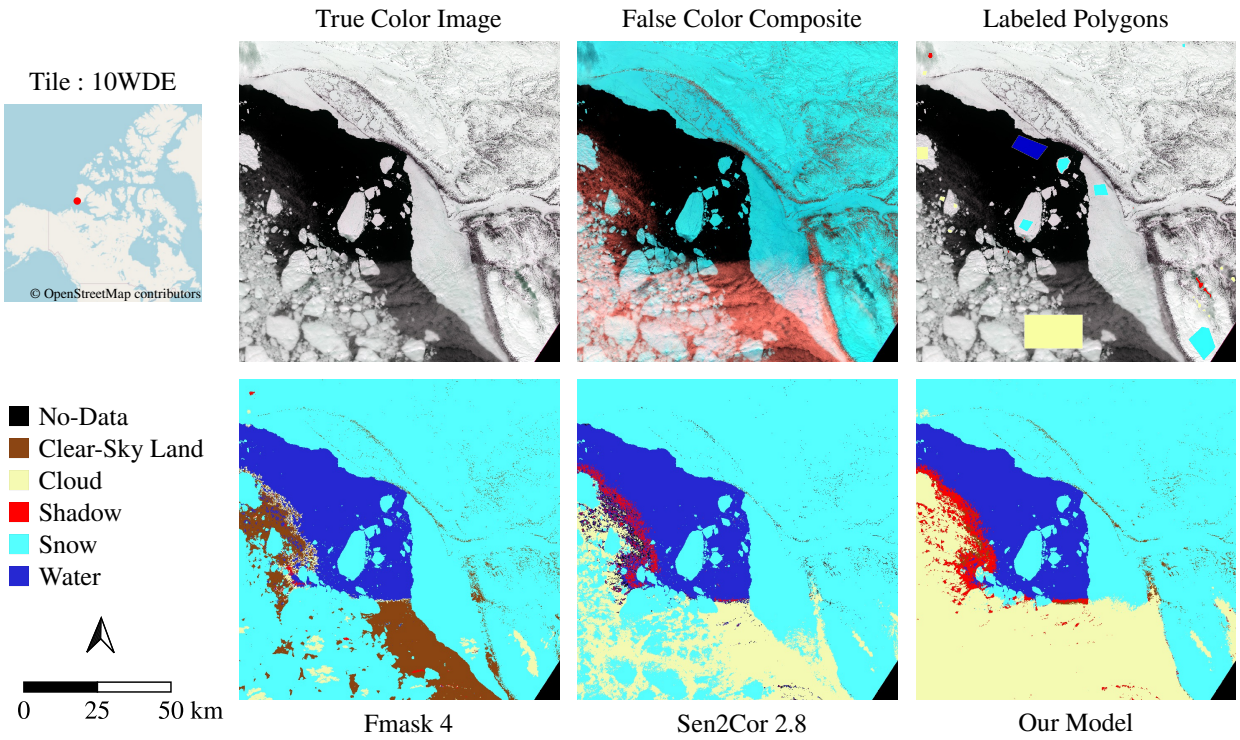

Figure 4. Comparison of test image results from Tile 10WDE in Northwest Territories, Canada, captured on 1 June 2020.

In Figures 5, A1, A2 and A3, we provide several more examples of results from the test and validation datasets to visually demonstrate the model performance. The results in each figure are supplemented with the false color composite image and the labeled polygons to aid in understanding the scene. The false color composite image comprises of the short-wave infrared band (B11), near infrared band (B08) and red band (B04) in the red, green and blue channels, respectively.

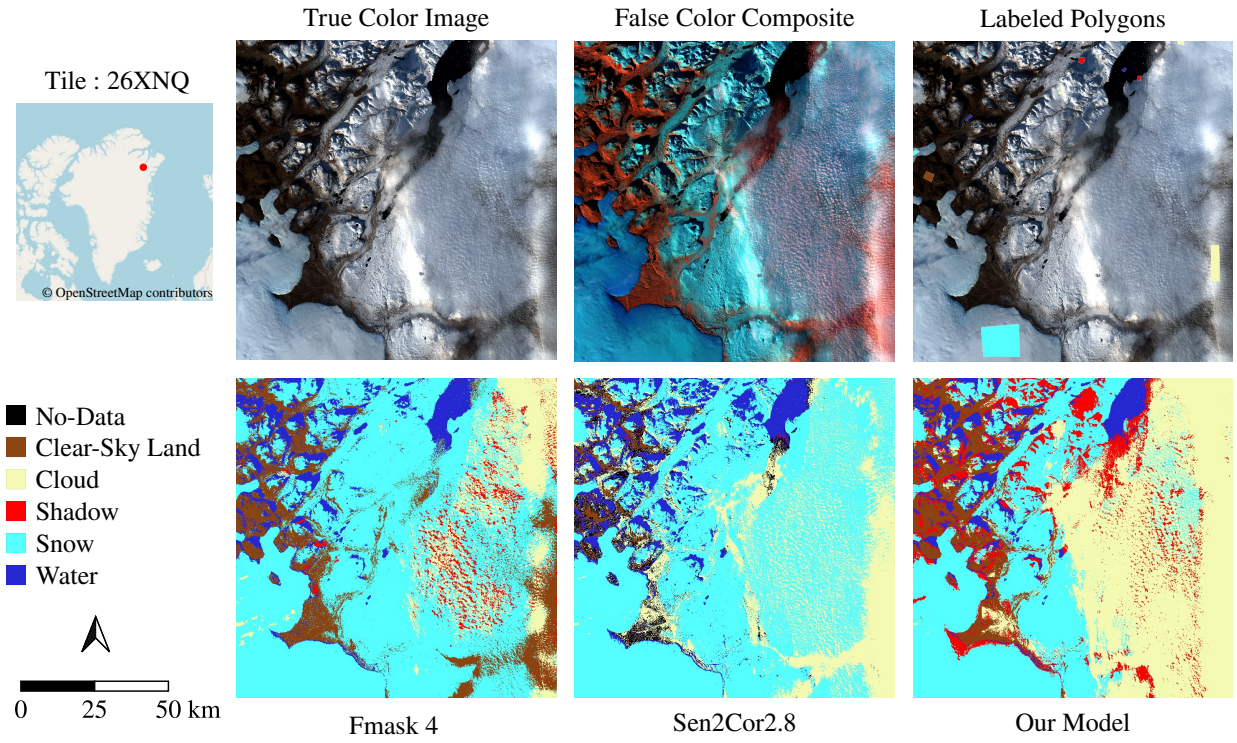

Figure 5. Comparison of test image results from Tile 26XNQ in North East Greenland, captured on 14 September 2020.

5. Discussion

In our study, we showed that, even with a small manually labeled validation dataset, the self-training framework enabled us to train a segmentation model using noisy labels from the Fmask algorithm. Our model outperformed two widely used cloud-screening methods, Sen2Cor and Fmask, and can be considered a better alternative to its teacher, the Fmask algorithm. The results showed that the model performed particularly well for the *Cloud* and *Snow* classes, which were the two prominent classes observed in the geographical sites that we used in our study.

Shadows in satellite imagery can be classified based on their source as cloud shadows or topographic shadows. Topographic shadows are the shadows that are cast by topographic features, such as mountains, and are static features that depend on acquisition geometry and solar position. In contrast, cloud shadows are dynamic features whose location and representation also depend on the prevailing meteorological conditions during image acquisition. The spectral characteristics of cloud shadows and topographic shadows are similar [48]. Topographic shadows can be detected accurately using digital elevation models (DEM) and solar angles. However this information is not provided to the network explicitly. The Fmask algorithm also makes use of the image metadata, such as solar zenith and azimuth angles, along with the cloud detections, in order to predict cloud shadow pixels. However, visual inspection of the Fmask results showed that they were not always very accurate. In Figure 5, both Sen2Cor and Fmask tended to incorrectly label topographic shadows as *Water*. Our model offered an improvement in this regard, correctly labelng them as shadow in many instances. A suitable post-processing technique using the metadata and DEM can be applied to separate the cloud shadows and topographic shadows.

In Table 10, we compare the performance metrics for different stages of the self-training framework. We observe that the performance of the model improved over the first three stages of the self-training framework and subsequently saturated in the last stage. We attribute this pattern to improvement in the quality of data labels as a result of our framework. The performance on the class *Water* did not improve across the different stages. We know that shadows and water tend to have dark-colored pixels and it is often difficult

to distinguish between them by visual inspection. We believe the network infers that the class *Water* and the class *Shadow* are similar, and, in cases of ambiguity, prefers *Shadow* over *Water* due to higher loss function weights used during training. As shown in Figure 6, the network focused on improving the performance in the shadow class at each stage and this improvement was also reflected in the performance metrics.

We also trained two models using the conventional supervised training approach. The first model was trained using the same training dataset, but using the noisy Fmask labels, while the second model was trained with the small human-labeled dataset that was previously used for model selection in our proposed framework. The network architecture used was the same as the architecture from the model in Stage-4; we also used all regularization techniques discussed previously for training this model. Hence, the key difference between the supervised models and the self-trained model (Stage-4) was the labels used for the training. We present the performance metrics for these two models and our self-trained model in Table 11. The results for the *clean* supervised model show that the limited number of labeled examples in our validation dataset was insufficient to generalize the task effectively. We believe that the large noisy dataset offered greater data diversity over the smaller clean dataset, and that his helped the *noisy* model to outperform the *clean* model. One might be inclined to argue that the improvement in performance across the stages of the self-training framework was due to the increase in the number of parameters used in the network architecture. However, the performance of the *noisy* supervised model demonstrated that a larger network alone does not guarantee better performance. We observed that the improvement in data labels achieved through a self-trained framework enabled the stage-2 and stage-3 models to outperform the supervised model despite their smaller network size. Hence, we believe our approach has great potential for training models with noisy labels.

Jeppesen et al. demonstrated the capability of neural networks to learn from noisy Fmask labels in their RS-Net model trained for cloud detection in Landsat Imagery using the supervised learning approach [28]. Our study can be viewed as a next step in this research direction, and we also further extend this approach for the multi-class segmentation task. Similar to our research, Li et al. offered a solution for addressing the difficulty of obtaining large pixel-level annotated datasets for training neural networks [36]. The block-level annotated dataset used for training their weakly supervised model can potentially simplify the annotation process in comparison to the more laborious pixel-level annotation. However, it may prove to be challenging to adapt this approach for multi-class segmentation, particularly for shadow detection. We used the simple, yet effective, U-Net architecture in our self-training framework. However we note that the network used in our framework can be easily adapted to take advantage of recent advances in network architectures and training techniques, which have shown improved cloud-detection capabilities in supervised models.

Table 10. Comparison of F1 score and IoU across the trained models from different stages of the self-training framework.

	Stage-1		**Stage-2**		**Stage-3**		**Stage-4**	
Class	**F1 Score**	**IoU**	**F1 Score**	**IoU**	**F1 Score**	**IoU**	**F1 Score**	**IoU**
Clear-Sky Land	0.61	0.44	0.81	0.68	0.92	0.84	0.92	0.85
Cloud	0.88	0.79	0.94	0.89	0.96	0.93	0.96	0.93
Shadow	0.68	0.52	0.76	0.62	0.81	0.67	0.81	0.68
Snow	0.83	0.71	0.91	0.83	0.93	0.86	0.93	0.87
Water	0.89	0.80	0.88	0.78	0.87	0.77	0.87	0.77
Total Accuracy	0.83		0.90		0.93		0.93	
mIoU	0.65		0.76		0.82		0.82	

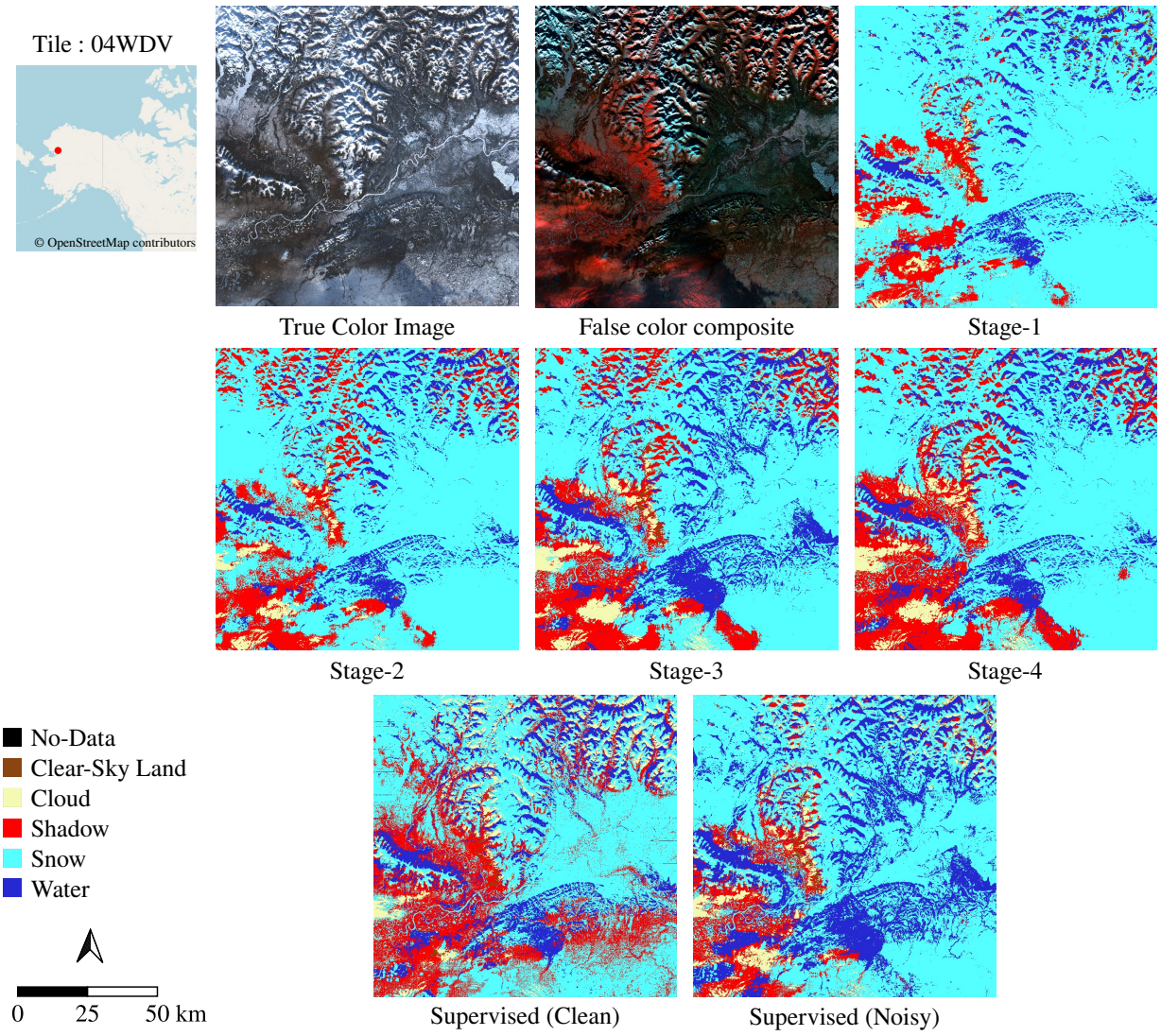

Figure 6. Comparison of test image results from different stages of the self-training framework and the results from the models trained using supervised training, from Tile 04WDV in Alaska, USA, captured on 27 October 2020.

Table 11. Comparison of F1 score and IoU with the models trained using the supervised training approach. The model in column Supervised (noisy) was trained on the entire training dataset using the Fmask labels and the model in column (clean) was trained on the validation dataset using the manually annotated labels.

	Our Model		**Supervised (Noisy)**		**Supervised (Clean)**	
Class	**F1 Score**	**IoU**	**F1 Score**	**IoU**	**F1 Score**	**IoU**
Clear-Sky Land	0.92	0.85	0.90	0.81	0.87	0.77
Cloud	0.96	0.93	0.95	0.90	0.94	0.88
Shadow	0.81	0.68	0.69	0.52	0.34	0.20
Snow	0.93	0.87	0.90	0.81	0.77	0.62
Water	0.87	0.77	0.82	0.69	0.85	0.74
Total Accuracy	0.93		0.89		0.84	
mIoU	0.82		0.75		0.64	

6. Conclusions

Our study demonstrates the effectiveness of self-training neural networks in the Earth observation domain. The key challenge in this study was to train the model using the noisy labels from the Fmask cloud detection algorithm. Many other remote-sensing applications face the same difficulty with existing, but imprecise, training data, that often limits the deployment of deep-learning technologies.

Even though the proposed method offers better performance compared to Fmask and Sen2Cor, we believe that the performance can be improved further. As well as the detection of shadows, the detection of thin clouds, particularly those above water bodies, can be improved. The clear-sky land class is the most heterogeneous of the six classes in this investigation. When training a similar model that can be applied to all geographical environments, this class is expected to pose some challenges. The use of other indices in addition to NDSI would allow advantage to be taken of the domain knowledge that has been acquired as a result of years of research.

Our classification strategy has the potential for more nuanced class separation and for integration into information retrieval algorithms. The use of other existing masks or data sets (e.g., ocean, lakes or vegetation) might lead to even higher precision on specific problems or over certain surfaces.

Author Contributions: Problem identification, M.H.B. and T.S.; conceptualization, V.I.M.; methodology, K.G.N. and V.I.M.; software, K.G.N.; validation, K.G.N., V.I.M. and P.H.; formal analysis, K.G.N. and V.I.M.; investigation, K.G.N.; resources, P.H.; data curation, K.G.N. and P.H.; writing—original draft preparation, K.G.N.; writing—review and editing, V.I.M.; visualization, K.G.N.; supervision, V.I.M. and P.H.; project administration, M.H.B. and T.S.; funding acquisition, V.I.M. and M.H.B. All authors have read and agreed to the published version of the manuscript.

Funding: This work was financially supported by the FAU Emerging Fields Initiative grant TAPE (Tapping the Potential of Earth Observation) and the STAEDLER Foundation. We also acknowledge financial support from Deutsche Forschungsgemeinschaft and Friedrich-Alexander-Universität Erlangen-Nürnberg within the funding program "Open Access Publication Funding".

Data Availability Statement: The code used for training the model is available at https://github.com/kmlnbr/deep-fmask (accessed on 25 February 2022). The labeled annotation files used in the validation and test dataset are available through PANGAEA, an international database hosted by the Alfred Wegener Institute, Helmholtz Center for Polar and Marine Research (AWI), and the Center for Marine Environmental Sciences, University of Bremen (MARUM), under https://doi.org/10.1594/PANGAEA.942321 (accessed on 25 February 2022).

Acknowledgments: We are grateful to the European Space Agency (ESA) for providing Sentinel-2 data through the Copernicus Data Hub. We also thank the OpenStreetMap contributors for providing the background maps used in our figures, under an Open Data Commons Open Database License (ODbL) (accessed on 25 February 2022).

Conflicts of Interest: The authors declare no conflict of interest. The funding sponsors had no role in the design of the study; in the collection, analysis, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Appendix A. Tile-Wise Quarterly Distribution of Sentinel-2 Scenes

Appendix A.1. Training Dataset

Table A1. Tile-wise quarterly distribution of scenes present in the training dataset. The size column indicates the count of 254 × 254 patches used for training.

Tile No.	Location	Jan.–Mar.	Apr.–Jun.	Jul.–Sep.	Oct.–Dec.	Size
06WVC	U.S.A	1		1	1	1390
07VCG	U.S.A		1		2	1452
11TLG	U.S.A	1	1		1	1452
11WNV	Canada		1	1	1	1452
12XVM	Canada	1		2		1452
14CMC	Antarctica	2			1	1452
15WXQ	Canada		1	2		1452
16CEU	Antarctica	1			2	1452
18DVF	Antarctica	2			1	1405
19DEE	Antarctica	1		1	1	1324
19JEH	Argentina	1	1		1	1452
19WER	Canada	1		2		1452
19XEH	Greenland	1		2		1348
20WMT	Canada		1	1	1	1452
20XNR	Greenland	1		2		1442
21CWJ	Antarctica	1			2	1121
21UUA	Canada	1			2	1452
21XWC	Greenland	1		1	1	1452
27XVB	Greenland	1		1	1	1452
27XWH	Greenland	1		2		1356
30XWR	Greenland Sea	1	1	1		1283
34WED	Norway			3		1448
42XVJ	Russia		1	1	1	1452
44XMF	Russia	1		2		1452
45DWG	Antarctica	1		1	1	1075
47XMJ	Russia		1	2		1452
49XDE	Russia	1		1	1	1452
54WVT	Russia	1	1	1		1452
55XDD	Russia	1		2		1368
58CDV	Antarctica	2			1	1452
59CMU	Antarctica	2		1		1452
60WWT	Russia		1	1	1	1446

Appendix A.2. Validation Dataset

Table A2. Tile-wise quarterly distribution of scenes present in the validation dataset.

Tile No.	Location	Jan.–Mar.	Apr.–Jun.	Jul.–Sep.	Oct.–Dec.	No. Labeled Pixels
16XEG	Canada	1		2		1,491,148
18CWU	Antarctica	1				678,120
19FDU	Chile				2	868,457
26XNR	Greenland		1	3		1,514,134
27XVL	Greenland			4		2,646,020
41XNE	Russia		1	1		913,888
45SVV	China			2	2	3,971,467
45WXR	Russia			1	1	1,064,095

Appendix A.3. Test Dataset

Table A3. Tile-wise quarterly distribution of scenes present in the test dataset.

Tile No.	Location	Jan.–Mar.	Apr.–Jun.	Jul.–Sep.	Oct.–Dec.	No. Labeled Pixels
04CEU	Antarctica	1			1	1,072,859
04WDV	U.S.A			1	2	590,574
10WDE	Canada		1	2		1,922,806
18XVM	Canada			2	1	1,438,918
26XNQ	Greenland			3		2,086,371
27XWK	Greenland			3		1,251,262
32VMP	Norway	1	1			764,357
42DVG	Antarctica	1				295,047
52XDF	Russia				3	2,174,747

Appendix B. Additional Image Results

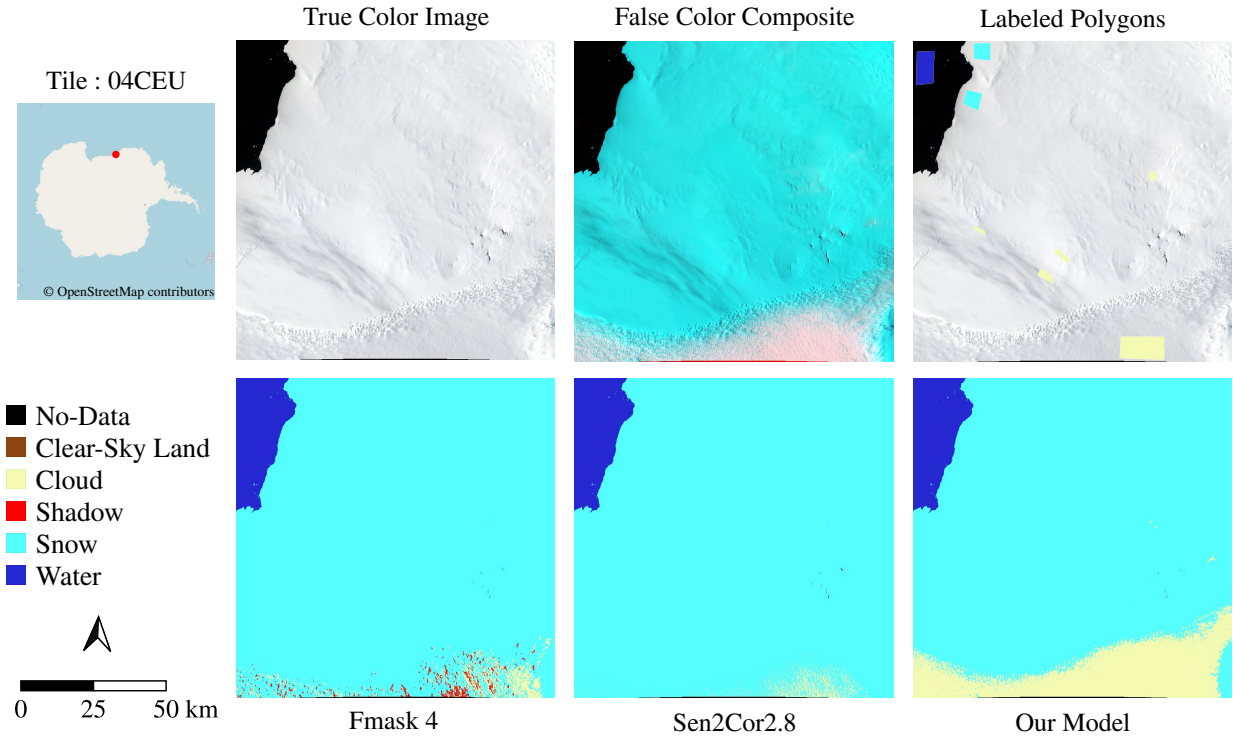

Figure A1. Comparison of test image results from Tile 04CEU in Marie Byrd Land, Antractica, captured on 12 December 2020.

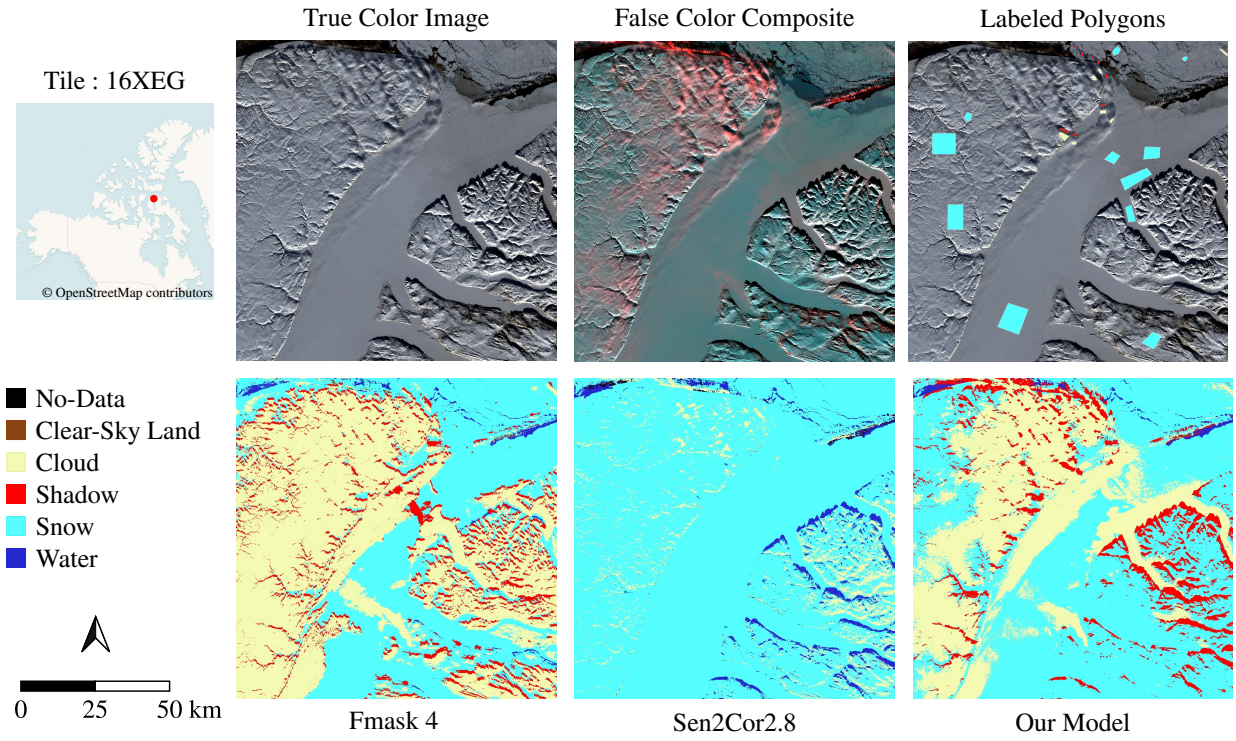

Figure A2. Comparison of validation image results from Tile 16XEG in Nunavut, Canada, captured on 15 March 2020.

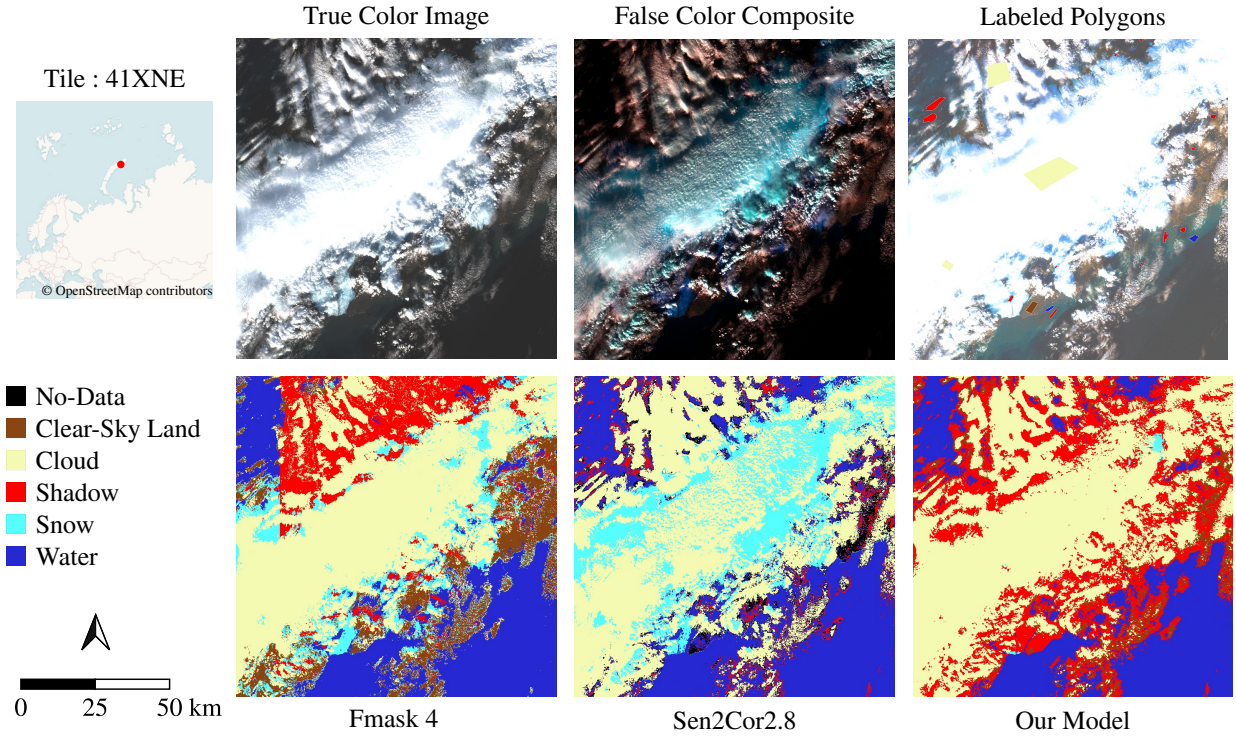

Figure A3. Comparison of validation image results from Tile 41XNE in Arkhangelsk Oblast, Russia, captured on 23 September 2020.

References

1. Zhu, Z.; Woodcock, C.E. Object-based cloud and cloud shadow detection in Landsat imagery. *Remote Sens. Environ.* **2012**, *118*, 83–94. [Google Scholar] [CrossRef].
2. Zhu, Z.; Wang, S.; Woodcock, C.E. 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.* **2015**, *159*, 269–277. [Google Scholar] [CrossRef].
3. Qiu, S.; Zhu, Z.; He, B. Fmask 4.0: Improved cloud and cloud shadow detection in Landsats 4–8 and Sentinel-2 imagery. *Remote Sens. Environ.* **2019**, *231*, 111205. [Google Scholar] [CrossRef].
4. Louis, J.; Debaecker, V.; Pflug, B.; Main-Knorn, M.; Bieniarz, J.; Mueller-Wilm, U.; Cadau, E.; Gascon, F. Sentinel-2 Sen2Cor: L2A processor for users. In Proceedings of the ESA Living Planet Symposium Living Planet Symposium, Prague, Czech Republic, 9–13 May 2016; pp. 1–8. [Google Scholar].
5. Christodoulou, C.I.; Michaelides, S.C.; Pattichis, C.S. Multifeature texture analysis for the classification of clouds in satellite imagery. *IEEE Trans. Geosci. Remote Sens.* **2003**, *41*, 2662–2668. [Google Scholar] [CrossRef].
6. Li, Z.; Shen, H.; Li, H.; Xia, G.; Gamba, P.; Zhang, L. Multi-feature combined cloud and cloud shadow detection in GaoFen-1 wide field of view imagery. *Remote Sens. Environ.* **2017**, *191*, 342–358. [Google Scholar][CrossRef].
7. Sun, L.; Wei, J.; Wang, J.; Mi, X.; Guo, Y.; Lv, Y.; Yang, Y.; Gan, P.; Zhou, X.; Jia, C.; et al. A universal dynamic threshold cloud detection algorithm (UDTCDA) supported by a prior surface reflectance database. *J. Geophys. Res. Atmos.* **2016**, *121*, 7172–7196. [Google Scholar] [CrossRef].
8. Zhou, G.; Zhou, X.; Yue, T.; Liu, Y. An optional threshold with SVM cloud detection algorithm and DSP implementation. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *XLI-B8*, 771–777. [Google Scholar] [CrossRef].
9. Sui, Y.; He, B.; Fu, T. Energy-based cloud detection in multispectral images based on the SVM technique. *Int. J. Remote Sens.* **2019**, *40*, 5530–5543. [Google Scholar] [CrossRef].
10. Hollstein, A.; Segl, K.; Guanter, L.; Brell, M.; Enesco, M. Ready-to-use methods for the detection of clouds, cirrus, snow, shadow, water and clear sky pixels in Sentinel-2 MSI images. *Remote Sens.* **2016**, *8*, 666. [Google Scholar] [CrossRef].
11. Ghasemian, N.; Akhoondzadeh, M. Introducing two random forest based methods for cloud detection in remote sensing images. *Adv. Space Res.* **2018**, *62*, 288–303. [Google Scholar] [CrossRef].
12. Le Hégarat-Mascle, S.; André, C. Use of Markov random fields for automatic cloud/shadow detection on high resolution optical images. *ISPRS J. Photogramm. Remote Sens.* **2009**, *64*, 351–366. [Google Scholar] [CrossRef].
13. Vivone, G.; Addesso, P.; Conte, R.; Longo, M.; Restaino, R. A class of cloud detection algorithms based on a MAP-MRF approach in space and time. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 5100–5115. [Google Scholar] [CrossRef].
14. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Munich, Germany, 5–9 October 2015; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2015; Volume 9351, pp. 234–241. [Google Scholar] [CrossRef].
15. Sallab, A.E.; Abdou, M.; Perot, E.; Yogamani, S. Deep reinforcement learning framework for autonomous driving. *Electron. Imaging* **2017**, *2017*, 70–76. [Google Scholar] [CrossRef].
16. Sohn, K.; Zhang, Z.; Li, C.L.; Zhang, H.; Lee, C.Y.; Pfister, T. A simple semi-supervised learning framework for object detection. *arXiv* **2020**, arXiv:2005.04757. [Google Scholar].
17. Zoph, B.; Ghiasi, G.; Lin, T.Y.; Cui, Y.; Liu, H.; Cubuk, E.D.; Le, Q. Rethinking pre-training and self-training. In *Advances in Neural Information Processing Systems*; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 3833–3845. [Google Scholar].
18. Xie, Q.; Luong, M.T.; Hovy, E.; Le, Q.V. Self-training with noisy student improves ImageNet classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 10687–10698. [Google Scholar] [CrossRef].
19. Lee, H.W.; Kim, N.r.; Lee, J.H. Deep neural network self-training based on unsupervised learning and dropout. *Int. J. Fuzzy Log. Intell. Syst.* **2017**, *17*, 1–9. [Google Scholar] [CrossRef].
20. Babakhin, Y.; Sanakoyeu, A.; Kitamura, H. Semi-supervised segmentation of salt bodies in seismic images using an ensemble of convolutional neural networks. In Proceedings of the German Conference on Pattern Recognition (GCPR), Dortmund, Germany, 10–13 September 2019; pp. 218–231. [Google Scholar] [CrossRef].
21. Chen, L.C.; Lopes, R.G.; Cheng, B.; Collins, M.D.; Cubuk, E.D.; Zoph, B.; Adam, H.; Shlens, J. Naive-student: Leveraging semi-supervised learning in video sequences for urban scene segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; pp. 695–714. [Google Scholar] [CrossRef].
22. Yilmaz, F.F.; Heckel, R. Image recognition from raw labels collected without annotators. *arXiv* **2019**, arXiv:1910.09055. [Google Scholar].
23. Huang, C.; Thomas, N.; Goward, S.N.; Masek, J.G.; Zhu, Z.; Townshend, J.R.G.; Vogelmann, J.E. Automated masking of cloud and cloud shadow for forest change analysis using Landsat images. *Int. J. Remote Sens.* **2010**, *31*, 5449–5464. [Google Scholar] [CrossRef].
24. Irish, R.R.; Barker, J.L.; Goward, S.N.; Arvidson, T. Characterization of the Landsat-7 ETM+ automated cloud-cover assessment (ACCA) algorithm. *Photogramm. Eng. Remote Sens.* **2006**, *72*, 1179–1188. [Google Scholar] [CrossRef].

25. Foga, S.; Scaramuzza, P.L.; Guo, S.; Zhu, Z.; Dilley, R.D., Jr.; Beckmann, T.; Schmidt, G.L.; Dwyer, J.L.; Hughes, M.J.; Laue, B. Cloud detection algorithm comparison and validation for operational Landsat data products. *Remote Sens. Environ.* **2017**, *194*, 379–390. [Google Scholar] [CrossRef].
26. Chai, D.; Newsam, S.; Zhang, H.K.; Qiu, Y.; Huang, J. Cloud and cloud shadow detection in Landsat imagery based on deep convolutional neural networks. *Remote Sens. Environ.* **2019**, *225*, 307–316. [Google Scholar] [CrossRef].
27. Xu, K.; Guan, K.; Peng, J.; Luo, Y.; Wang, S. DeepMask: An algorithm for cloud and cloud shadow detection in optical satellite remote sensing images using deep residual network. *arXiv* **2019**, arXiv:1911.03607. [Google Scholar].
28. Jeppesen, J.H.; Jacobsen, R.H.; Inceoglu, F.; Toftegaard, T.S. A cloud detection algorithm for satellite imagery based on deep learning. *Remote Sens. Environ.* **2019**, *229*, 247–259. [Google Scholar] [CrossRef].
29. Mohajerani, S.; Saeedi, P. Cloud-Net: An end-to-end cloud detection algorithm for Landsat 8 imagery. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Yokohama, Japan, 28 July–2 August 2019; pp. 1029–1032. [Google Scholar] [CrossRef].
30. Shao, Z.; Pan, Y.; Diao, C.; Cai, J. Cloud detection in remote sensing images based on multiscale features-convolutional neural network. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 4062–4076. [Google Scholar] [CrossRef].
31. Zhan, Y.; Wang, J.; Shi, J.; Cheng, G.; Yao, L.; Sun, W. Distinguishing cloud and snow in satellite images via deep convolutional network. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1785–1789. [Google Scholar][CrossRef].
32. Yan, Z.; Yan, M.; Sun, H.; Fu, K.; Hong, J.; Sun, J.; Zhang, Y.; Sun, X. Cloud and cloud shadow detection using multilevel feature fused segmentation network. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 1600–1604. [Google Scholar][CrossRef].
33. Zhang, L.; Sun, J.; Yang, X.; Jiang, R.; Ye, Q. Improving deep learning-based cloud detection for satellite images with attention mechanism. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5. [Google Scholar] [CrossRef].
34. Yu, J.; Li, Y.; Zheng, X.; Zhong, Y.; He, P. An effective cloud detection method for Gaofen-5 images via deep learning. *Remote Sens.* **2020**, *12*, 2106. [Google Scholar] [CrossRef].
35. Liu, Y.; Wang, W.; Li, Q.; Min, M.; Yao, Z. DCNet: A deformable convolutional cloud detection network for remote sensing imagery. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5. [Google Scholar] [CrossRef].
36. Li, Y.; Chen, W.; Zhang, Y.; Tao, C.; Xiao, R.; Tan, Y. Accurate cloud detection in high-resolution remote sensing imagery by weakly supervised deep learning. *Remote Sens. Environ.* **2020**, *250*, 112045. [Google Scholar] [CrossRef].
37. Liu, C.C.; Zhang, Y.C.; Chen, P.Y.; Lai, C.C.; Chen, Y.H.; Cheng, J.H.; Ko, M.H. Clouds classification from Sentinel-2 imagery with deep residual learning and semantic image segmentation. *Remote Sens.* **2019**, *11*, 119. [Google Scholar] [CrossRef].
38. Li, J.; Wu, Z.; Hu, Z.; Jian, C.; Luo, S.; Mou, L.; Zhu, X.X.; Molinier, M. A lightweight deep learning-based cloud detection method for Sentinel-2A imagery fusing multiscale spectral and spatial features. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–19. [Google Scholar] [CrossRef].
39. Hughes, M.J.; Kennedy, R. High-quality cloud masking of Landsat 8 imagery using convolutional neural networks. *Remote Sens.* **2019**, *11*, 2591. [Google Scholar] [CrossRef].
40. ESA. Sentinel-2 Spectral Band Information. Available online: https://sentinel.esa.int/web/sentinel/user-guides/sentinel-2-msi/resolutions/radiometric (accessed on 19 June 2020).
41. QGIS Development Team. *QGIS Geographic Information System*; QGIS Association: Grüt, Switzerland, 2021.
42. Qiu, S.; He, B.; Zhu, Z.; Liao, Z.; Quan, X. Improving Fmask cloud and cloud shadow detection in mountainous area for Landsats 4–8 images. *Remote Sens. Environ.* **2017**, *199*, 107–119. [Google Scholar] [CrossRef].
43. Hall, D.K.; Riggs, G.A.; Salomonson, V.V. Development of methods for mapping global snow cover using moderate resolution imaging spectroradiometer data. *Remote Sens. Environ.* **1995**, *54*, 127–140. [Google Scholar] [CrossRef].
44. DeVries, T.; Taylor, G.W. Improved regularization of convolutional neural networks with cutout. *arXiv* **2017**, arXiv:1708.04552. [Google Scholar].
45. Tompson, J.; Goroshin, R.; Jain, A.; LeCun, Y.; Bregler, C. Efficient object localization using convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 648–656. [Google Scholar] [CrossRef].
46. Eigen, D.; Fergus, R. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 11–18 December 2015; pp. 2650–2658. [Google Scholar] [CrossRef].
47. Kampffmeyer, M.; Salberg, A.B.; Jenssen, R. Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Las Vegas, NV, USA, 29 June–1 July 2016; pp. 680–688. [Google Scholar] [CrossRef].
48. Martinuzzi, S.; Gould, W.A.; González, O.M.R. *Creating Cloud-Free Landsat ETM+ Data Sets in Tropical Landscapes: Cloud and Cloud-Shadow Removal*; General Technical Report IITF-32; US Department of Agriculture, Forest Service, International Institute of Tropical Forestry: Rio Piedras, PR, USA, 2007. [Google Scholar] [CrossRef].